AD-A067 979 NAVAL UNDERWATER SYSTEMS CENTER NEW LONDON CONN NEW --ETC F/G 17/1
ARRAY OPTIMIZATION USING SUBARRAYS.(U)
MAR 79 R L STREIT
UNCLASSIFIED NUSC-TR-5889 NL

1 OF 1
AD
A067979

END
DATE
FILMED
6-79
DDC

AD A067979

DDC FILE COPY

NUSC Technical Report 5889

(12)

LEVEL II

Array Optimization Using Subarrays

Roy L. Streit
Computer Sciences Department

D D C
RECEIVED
MAY 1 1979
A

23 March 1979

NUSC
Naval Underwater Systems Center
Newport, Rhode Island • New London, Connecticut

Approved for public release; distribution unlimited.

79 04 27 003

PREFACE

The research reported herein was conducted under NUSC Project No. A70270, "Optimal Synthesis of Three Dimensional Arrays," Principal Investigator, R. L. Streit (Code 4424).

The Technical Reviewer for this report was B. G. Buehler (Code 3103). The author would like to thank Mr. Buehler for the use of one of his examples.

REVIEWED AND APPROVED: 23 March 1979

D L. Nichols

D. L. Nichols
Associate Technical Director
for Engineering and Technical Support

The author of this report is located at the New London
Laboratory, Naval Underwater Systems Center,
New London, Connecticut 06320.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TR 5889 ✓

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 ARRAY OPTIMIZATION USING SUBARRAYS.

5. TYPE OF REPORT & PERIOD COVERED
9 Technical rept.,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 Roy L. Streit

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Underwater Systems Center
New London Laboratory
New London, CT 06320

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
A60040 12 58p.

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Underwater Systems Center
Newport, RI 02840

12. REPORT DATE
11 23 March 1979

13. NUMBER OF PAGES
55

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
14 NUSC-TR-5889

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Antenna Arrays
Antenna Subarrays
Beamwidth Control (BC)
Directivity Index with Beamwidth Control (DIBC)
Optimization of Discrete Arrays
Group Coordinate Relaxation
Maximization Techniques

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The concept of directivity index with beamwidth control (DIBC) leads to a practical method for the selection, or optimization, of element excitations to control beamwidth and sidelobe level in a discrete array of arbitrary configuration. This optimization procedure depends upon the design frequency, the specified element positions, individual element field patterns, and the ambient noise field. Each of these factors can be specified in a completely general manner. In addition, the optimization procedure may be adapted to computers

(over)

405 918

LB next page

20. (Cont'd)

of modest memory size. Examples are included to show the versatility of this approach to the optimization problem, as well as its limitations.

TABLE OF CONTENTS

Page

ACCESSION for
NTIS White Section ☒
DDC Buff Section □
UNANNOUNCED □
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

79 04 27 003

LIST OF ILLUSTRATIONS

LIST OF TABLES

Array Optimization Using Subarrays

THE CONCEPT

INTRODUCTION

Optimization of the element excitations of discrete antenna arrays is a matter of definition for three reasons. First, the definition of optimality will dictate the appropriate mathematical approach. Seemingly subtle changes in the definition of optimality can radically alter the applicable mathematical methods. Second, element excitations that are optimal in one sense are unlikely to be optimal in another sense. Two sets of excitations, each set optimal in its own sense, can be completely different. Third, the definition of optimality must reflect directly upon the primary design goals for the array. It does no good to optimize the directivity index and then complain that the sidelobes are too high because the design goal of low sidelobes and the definition of optimality (maximum DI) are not directly related.

This report defines and uses exclusively the concept of directivity index with beamwidth control (DIBC). Several advantages as well as difficulties inherent in this definition are discussed. The primary difficulty in this definition is the requirement of large computer memories for large arrays. A technique employing subarrays of the full array in a systematic manner is shown to overcome this problem.

The optimization procedure in this report is applicable when the following premises obtain:

1. The wavelength λ of the design frequency is given and fixed.

2. The number n of elements in the array is fixed and all the element positions (x_k, y_k, z_k), $k = 1,\ldots,n$ are known and fixed.

3. Individual element field patterns at the design frequency are completely known.

4. The ambient noise field at the design frequency is completely known.

5. Element interactions can be ignored.

6. Element excitations can be phased (i.e., complex).

FIELD PATTERNS AND COORDINATE SYSTEM

The spherical coordinate system of figure 1 is used throughout this report; however, a particular direction (θ,ϕ) will be specified by the direction cosines

$$\begin{aligned} \cos\alpha &= \sin\phi\,\cos\theta \\ \cos\beta &= \sin\phi\,\sin\theta \\ \cos\gamma &= \cos\phi\;. \end{aligned} \tag{1}$$

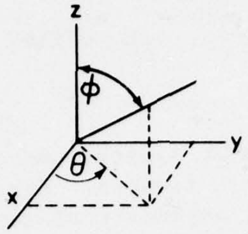

Figure 1. The Coordinate System

The most general field pattern treated here is

$$V(\theta,\phi) = \sum_{k=1}^{n} a_k\, R_k(\theta,\phi)\, \exp\left\{\frac{2\pi i}{\lambda}\, d_k(\theta,\phi)\right\}, \tag{2}$$

where $R_k(\theta,\phi)$ is the phased response of the k-th element, and

$$d_k(\theta,\phi) = x_k \cos\alpha + y_k \cos\beta + z_k \cos\gamma\;. \tag{3}$$

Because of assumptions 1-6, the field pattern $V(\theta,\phi)$ depends solely upon the phased (complex) excitations $a_1,\ldots,a_n$. The ambient noise field $N(\theta,\phi)$ enters only in the definition of optimal excitations.

DIRECTIVITY INDEX WITH BEAMWIDTH CONTROL (DIBC)

The antenna designer is required to divide the set of all directions, denoted Ω, into three disjoint regions:

$\mathcal{M}$ = Mainlobe region
$\mathcal{S}$ = Sidelobe region
$\mathcal{I}$ = Ignored region = $\Omega - (\mathcal{M} \cup \mathcal{S})$.

This division of directional space is completely arbitrary, except that neither $\mathcal{M}$ nor $\mathcal{S}$ can be empty sets, whereas $\mathcal{I}$ can be empty if desired. Once a particular choice of $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{I}$ has been made, the following definition of optimality is used.

Definition 1. The element excitations $a_1,\ldots,a_n$ are optimal excitations for a given choice of regions $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{I}$, if and only if the ratio

$$\mathrm{DIBC} = \frac{\iint_{\mathcal{M}} \left|N(\theta,\phi)\; V^2(\theta,\phi)\right| \sin\phi\; d\phi d\theta}{\iint_{\mathcal{M}\cup\mathcal{S}} \left|N(\theta,\phi)\; V^2(\theta,\phi)\right| \sin\phi\; d\phi d\theta} \tag{4}$$

is maximized. Any ratio of this form will be referred to as a Directivity Index with Beamwidth Control (DIBC).

Maximizing Directivity Index (DI) is a limiting case of maximizing DIBC. To see this, recall that for a specified direction (θ_0, ϕ_0), DI is a maximum if the ratio

$$DI = \frac{\left| N(\theta_0, \phi_0)\, V^2(\theta_0, \phi_0) \right|}{\iint_{\Omega} \left| N(\theta, \phi)\, V^2(\theta, \phi) \right| \sin\phi \, d\phi d\theta} \tag{5}$$

is maximized. Now let the ignored region $\mathcal{I}$ be empty, let the mainlobe region $\mathcal{M}$ contain (θ_0, ϕ_0), and let $\mathcal{S} = \Omega - \mathcal{M}$. Then excitations maximizing DIBC converge to excitations which maximize DI as the mainlobe region $\mathcal{M}$ shrinks down on the point (θ_0, ϕ_0).

We have defined optimal excitations as those for which DIBC is maximized for some choice of regions $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{I}$. This allows a measure of control over the beamwidth and sidelobe level. By varying systematically the choice of $\mathcal{M}$ and $\mathcal{S}$, and with each choice maximizing the DIBC, we can directly examine the tradeoff between beamwidth and sidelobe level for the particular array at hand. The engineer can then select those excitations which best suit his needs. Generally, the larger the mainlobe region $\mathcal{M}$ and the smaller the sidelobe region $\mathcal{S}$ (for fixed ignored region $\mathcal{I}$), the lower the overall sidelobe level and the greater the beamwidth. However, this is not always the case, since sidelobe level does not enter directly into the DIBC ratio of equation (4). Nothing prevents the field pattern from having narrow high amplitude sidelobes, since such sidelobes would contribute little to the denominator of the DIBC.

Another reason for maximizing DIBC is simply that it is conceptually easy to do so. All that is required is the solution of an eigenvalue/eigenvector problem (see Theorem 1), and problems of this type have been extensively studied in the literature (reference 1). Numerically, such problems require considerable care. Fortunately, well-designed computer programs are available for the solution of eigenproblems (references 2 and 3). With the use of these routines, the solutions of the eigenproblems encountered in the antenna problem seem to be numerically stable. This is not to say that there may not be arrays which yield numerically unstable eigenproblems.

A final reason for maximizing DIBC is more esoteric. In the process of solving the required eigenproblem, all the eigenvalue/eigenvector pairs can be computed, not merely the largest one. It happens that the field patterns corresponding to the lower order eigenvalues have some interesting features (see example 1, figures 5-8). In addition, it happens that the larger eigenvalues are "close together"; i.e., several linearly independent sets of excitations exist which give DIBC values that lie "close together." (For an analogous situation, see reference 4.) What this means in the antenna problem is that without sacrificing antenna performance (as measured solely by the DIBC), it becomes a simple matter to examine numerous different sets of excitations with the aim of improving some completely different design goal of the array. This will not be discussed further in this report.

It must be mentioned that this approach to the array optimization problem does not attempt to address several issues which are of practical interest. First, this approach does not guarantee that the array performance is insensitive to perturbations in the optimum excitations. The question of sensitivity to excitation perturbation can be examined only after the optimum excitations are found. Second, this approach does not attempt to control the efficiency of the array. In other words, it can happen that the optimal excitations for a particular array may drive certain elements at their maximum allowed levels while the remaining elements are hardly driven at all, so that the total power out of the array is too low for the application. This problem is common to all amplitude shaded arrays and can be examined after the optimum excitations are found. Finally, this approach to array optimization requires that the optimal excitations be allowed to be complex (i.e., phased), and this can require certain elements to absorb power from the medium. We stress that this, as well as the other two practical matters, must be addressed after the optimal excitations are found.

COMPUTER STORAGE PROBLEM

The primary drawback to maximizing DIBC is that the number of computer storage locations required (using the program in appendix B) is approximately

$$N_T = 6n^2 + 16n + 12,000 \quad , \qquad (6)$$

for the case of constant ambient noise field and omnidirectional elements, (table 1). Since the total requirement will grow as the ambient noise field and/or element field patterns require more storage to compute, it appears that the direct computation of optimal excitations for any array of 100 or more elements requires large, main frame computers.

Table 1. Storage Requirements for Direct Computation of Maximum DIBC

No. of Elements, n	Minimum Storage Required, N_T
15	13,590
30	17,880
60	34,560
100	73,600
150	149,400
200	255,200

The storage requirements for maximizing DIBC can be avoided. A technique known as group coordinate relaxation (reference 5) gives a method which can be tailored to the amount of computer memory available. The technique is an excellent example of how to trade off computer memory for computational speed. The more memory available, the faster the DIBC can be maximized.

Group coordinate relaxation, in the context of maximizing DIBC, is simply stated. Suppose there are 300 elements in the array. Make any initial guess at the optimal excitations. Define distinct subarrays of, say, 50 elements each. By working with the first of these subarrays, new element excitations are computed for these 50 elements, so that the DIBC of the entire 300 element array is increased. Next, new excitations are computed for the second subarray. Cycling through all six subarrays in turn, until DIBC for the entire 300 element array cannot be increased further by changing the excitations in any of the subarrays, is the essence of group coordinate relaxation. The method can be proved to be convergent. The rate of convergence depends heavily on the size of the subarrays used. The larger the subarrays, the faster the convergence, and the more core storage required. Thus, core storage is traded off in a direct manner for the convergence rate, and hence for computation time. In addition, each step of the group coordinate relaxation method produces new excitations which increase the DIBC, so that if the computations are interrupted for any reason:

(1) The last computed excitations are better than any of the excitations previously computed,

(2) By saving the last computed excitations, the computations can be resumed without any significant loss.

If n_s is the number of elements in a subarray used by the group coordinate relaxation process, the total storage required (using the program in appendix B) is approximately

$$N_R = 6n_s^2 + 8(n + n_s) + 12{,}000 \ , \qquad (7)$$

for the case of constant ambient noise field and omnidirectional elements (table 2).

Table 2. Storage Requirements for Maximizing DIBC of 200-Element Array by Group Coordinate Relaxation

No. of Subarray Elements, n_s	Minimum Storage Required, N_R
15	15,070
30	19,240
60	35,680
100	74,400
150	149,800
200	255,200

ELABORATION OF THE CONCEPT

DIBC AND THE EIGENPROBLEM

Let the vector $a = \langle a_1,\ldots,a_n\rangle^T$ be the vector of element excitations for the field pattern $V(\theta,\phi)$ given by equation (2). Then

$$\iint_{\mathcal{M}} |N(\theta,\phi)\ V^2(\theta,\phi)|\ \sin\phi\ d\phi d\theta$$

$$= \iint_{\mathcal{M}} |N(\theta,\phi)|\ |V(\theta,\phi)|^2\ \sin\phi\ d\phi d\theta$$

$$= \iint_{\mathcal{M}} |N(\theta,\phi)|\ \overline{\left\{\sum_{k=1}^{n} a_k R_k(\theta,\phi) \exp\left[\frac{2\pi i}{\lambda}\ d_k(\theta,\phi)\right]\right\}}$$

$$\cdot \left\{\sum_{j=1}^{n} a_j R_j(\theta,\phi) \exp\left[\frac{2\pi i}{\lambda}\ d_j(\theta,\phi)\right]\right\} \sin\phi\ d\phi d\theta$$

$$= \sum_{k=1}^{n}\ \sum_{j=1}^{n} \overline{a}_k a_j \iint_{\mathcal{M}} |N(\theta,\phi)|\ \overline{R_k(\theta,\phi)}\ R_j(\theta,\phi)$$

$$\cdot\ \exp\left\{\frac{2\pi i}{\lambda}\left[d_j(\theta,\phi) - d_k(\theta,\phi)\right]\right\} \sin\phi\ d\phi d\theta$$

$$= \overline{a}^T U a,$$

where U is an n x n complex matrix. If $U = [u_{kj}]$ with k denoting the row number and j denoting the column number, then

$$u_{kj} = \iint_{\mathcal{M}} |N(\theta,\phi)|\ R_j(\theta,\phi)\overline{R_k(\theta,\phi)} \exp\left\{\frac{2\pi i}{\lambda}\left[d_j(\theta,\phi) - d_k(\theta,\phi)\right]\right\} \sin\phi\ d\phi d\theta. \qquad (8)$$

Clearly, U is a Hermitian matrix (i.e., $U = \overline{U}^T$), since it is obvious that $u_{kj} = \overline{u}_{jk}$. Also, U is positive definite since

$$\overline{a}^T U a \equiv \iint_{\mathcal{M}} |N(\theta,\phi)\ V^2(\theta,\phi)|\ \sin\phi\ d\phi d\theta > 0\ , \qquad (9)$$

whenever the excitation vector a ≠ 0 (and provided the mainlobe region $\mathcal{M}$ is not a set of measure zero, a pathological condition that is not encountered in this application). Therefore, for every mainlobe region $\mathcal{M}$, the matrix U defined in equation (8) is an n x n positive definite Hermitian matrix. Similarly,

$$\iint_{\mathcal{M}\cup\mathcal{S}} |N(\theta,\phi)\ V^2(\theta,\phi)| \sin\phi\ d\phi d\theta = \overline{a}^T W a\quad ,$$

where $W = [w_{kj}]$ is an n x n positive definite Hermitian matrix whose general entry is

$$w_{kj} = \iint_{\mathcal{M} \cup \mathcal{S}} |N(\theta,\phi)| \; R_j(\theta,\phi) \; \overline{R_k(\theta,\phi)} \exp\left\{\frac{2\pi i}{\lambda}\left[d_j(\theta,\phi)-d_k(\theta,\phi)\right]\right\} \sin\phi \; d\phi d\theta. \tag{10}$$

Thus, for a given choice of $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{L}$, we have

$$\text{DIBC} = \frac{\bar{a}^T U a}{\bar{a}^T W a} \quad , \tag{11}$$

which is a ratio of positive definite Hermitian forms. Therefore, optimal excitations are those that maximize this ratio of Hermitian forms.

The mathematical tools for handling ratios of the form of equation (11) have been known for at least a century. We have the following general mathematical result.

Theorem 1. If U and W are n x n Hermitian matrices and W is positive definite, then the eigenvalues of the generalized eigenproblem

$$Uz = \mu W z \tag{12}$$

are all real. Let $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$ denote these eigenvalues. Then linearly independent vectors $z_1,\ldots,z_n$ can be found which satisfy

$$U z_k = \mu_k W z_k \; , \quad k = 1,\ldots,n \tag{13}$$

and

$$\bar{z}_k^T W z_j = \begin{cases} 1 \text{ if } k = j \\ 0 \text{ if } k \neq j. \end{cases} \tag{14}$$

The vectors $z_1,\ldots,z_n$ are called the eigenvectors of the eigenproblem (12). Also, we have

$$\max_{z \neq 0} \left\{ \frac{\bar{z}^T U z}{\bar{z}^T W z} \right\} = \mu_1 \quad , \tag{15}$$

and this maximum is attained for every eigenvector corresponding to μ_1, and

$$\min_{z \neq 0} \left\{ \frac{\bar{z}^T U z}{\bar{z}^T W z} \right\} = \mu_n , \tag{16}$$

and this minimum is attained for every eigenvector corresponding to μ_n. Finally, if $1 \leq k \leq n$, then for any constants $\alpha_1,\ldots,\alpha_k$ not all zero, we have

$$\mu_1 \geq \frac{\bar{z}^T U z}{\bar{z}^T W z} \geq \mu_k \;, \tag{17}$$

where $z = \alpha_1 z_1 + \ldots + \alpha_k z_k$.

The proofs of the various parts of this theorem can be found in numerous sources, e.g., reference 1.

For the immediate purposes, the most important part of this theorem is equation (15). It states that optimal excitations are precisely the components of any eigenvector corresponding to the largest eigenvalue of the generalized eigenproblem $Uz = \mu Wz$, where U and W are defined by equations (8) and (10).

Theoretically, Theorem 1 solves the problem of maximizing DIBC completely. A discrete reformulation of DIBC is discussed next, followed by a discussion of the numerical methods for the solution of the required eigenproblem.

A DISCRETE VERSION OF DIBC

Maximizing the DIBC ratio (4) is mathematically tractable, but it is not practical. It requires the solution of an eigenproblem, which in turn requires the evaluation of approximately n^2 double integrals over subsets of the unit sphere. Since it is essential that the mainlobe region $\mathcal{M}$ and the sidelobe region $\mathcal{S}$ be quite general in nature (i.e., be defined to suit the particular situation), these double integrals are impossible to evaluate explicitly in general, and are also difficult and time consuming to evaluate accurately by numerical methods. For these reasons, DIBC itself is NOT optimized. What is optimized is a discrete version (DIBCF_p) of DIBC that is not only numerically practical to use, but is also conceptually simple.

The discrete DIBC definition replaces the surface integrals in ratio (4) by discrete sums over points chosen in $\mathcal{M}$ and $\mathcal{S}$. Since $\mathcal{M}$ and $\mathcal{S}$ are not known a priori, these points are distributed "uniformly" over the surface of the sphere, with each point contributing one term to the discrete sum, and all terms entering with equal weight. Ideally, then, these points must show no directional bias and must be easy to compute. Furthermore, it must be possible to choose these points with any desired "density" on the sphere.

A natural choice for points fulfilling these conditions is easy to describe, but difficult to compute: Choose as points the equilibrium positions of a finite number of positive charges constrained to lie on the surface of the unit sphere. When the number of positive charges is 4, 6, 8, 12, or 20, it is intuitively clear that stable points for these charges are at the vertices of the 5 regular Platonic bodies: the tetrahedron, the octahedron, the cube, the icosahedron, and the dodecahedron, respectively. Unfortunately, these are the only easy cases (see reference 6).

The discrete points chosen to define discrete DIBC are the vertices of a geodesic dome. Consider the icosahedron shown in figure 2.

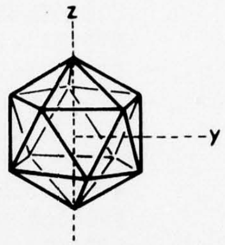

Figure 2. The Icosahedron

(Note that in this figure the y-axis is in the plane of the paper and the z-axis is tilted slightly to show off the configuration. The x-axis is not shown, but is, of course, orthogonal to the yz-plane.) This regular figure has 12 vertices, 20 faces, and 30 edges. Geodesic domes with (almost) any number of faces are constructed from the icosahedron by subdividing its equilateral triangular faces in a systematic manner (reference 7). First, subdivide each face into congruent equilateral subtriangles as shown in figure 3; that is, for each positive integer $p \geq 1$, find p+1 equispaced points along each edge and pass lines through each of these points parallel to the other two edges. Next, take all the vertices of the equilateral subtriangles so generated and project them onto the unit sphere. By doing this for each face of the icosahedron for a fixed integer $p \geq 1$, we construct the vertices of a geodesic dome of order p. We define the Fuller points $\mathfrak{F}_p$ to be the totality of these points.

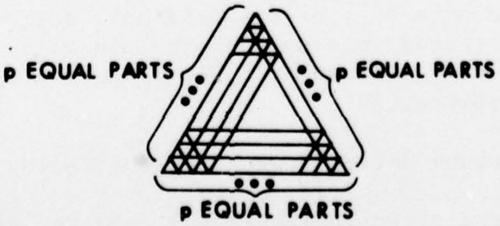

Figure 3. One Face of Icosahedron Subdivided into p Parts

The Fuller points $\mathfrak{I}_p$ are uniquely oriented in Cartesian space once the vertices of the icosahedron are defined. With some simple trigonometry, table 3 results. How many points are there in $\mathfrak{I}_p$? By inspecting an "unfolded" paper model of the icosahedron, it is easy to see that $\mathfrak{I}p$, contains exactly 10 p^2 + 2 points. Table 4 gives an idea of the density of these points on the unit sphere.

Table 3. Vertices of the Icosahedron

x	y	z
.000000000	.000000000	1.000000000
.894427191	.000000000	.447213595
.276393202	.850650808	.447213595
-.723606798	.525731112	.447213595
-.723606798	-.525731112	.447213595
.276393202	-.850650808	.447213595
.723606798	.525731112	- .447213595
-.276393202	.850650808	- .447213595
-.894427191	.000000000	- .447213595
-.276393202	-.850650808	- .447213595
.723606798	-.525731112	- .447213595
.000000000	.000000000	-1.000000000

Table 4. Number and Density of Fuller Points $\mathfrak{I}_p$

p	No. of Points in $\mathfrak{I}_p$	No. of Steradians/Point
1	12	1.047
2	42	.299
4	162	.0776
8	642	.0196
16	2562	.00490
24	5762	.00218
32	10,242	.00123

Notice that the Fuller points $\mathfrak{I}_p$ are not quite ideal. Those points chosen near the center of a face of the original icosahedron will be less finely spaced when projected onto the sphere than will those points that were chosen nearer an edge. This defect in $\mathfrak{I}_p$ does not seem to be significant in this application.

With the Fuller points defined, we state the following.

Definition 2. For a given integer $p \geq 1$, and regions $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{L}$, the element excitations $a_1,\ldots,a_n$ are optimal if and only if the ratio

$$\mathrm{DIBCF}_p = \frac{\dfrac{1}{|\mathcal{F}_p \cap \mathcal{M}|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap\mathcal{M}} |N(\theta,\phi)v^2(\theta,\phi)|\right\}}{\dfrac{1}{|\mathcal{F}_p \cap (\mathcal{M}\cup\mathcal{S})|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap(\mathcal{M}\cup\mathcal{S})} |N(\theta,\phi)\ v^2(\theta,\phi)|\right\}} \qquad (18)$$

is maximized, where $|\mathcal{F}_p \cap \mathcal{M}|$ is the number of points of $\mathcal{F}_p$ that lie in $\mathcal{M}$ and $|\mathcal{F}_p \cap (\mathcal{M}\cup\mathcal{S})|$ is the number of points of $\mathcal{F}_p$ that lie in $\mathcal{M}\cup\mathcal{S}$. Any ratio of the form of equation (18) will be referred to as a Directivity Index with Beamwidth Control over the Fuller points $\mathcal{F}_p$ (DIBCF_p).

The formulation of DIBCF_p as an eigenproblem parallels that for DIBC. Specifically, we have

$$Mz = \mu Sz \ , \qquad (19)$$

where $M = [m_{kj}]$ and $S = [s_{kj}]$ are n x n positive definite Hermitian matrices with

$$m_{kj} = \frac{1}{|\mathcal{F}_p \cap \mathcal{M}|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap\mathcal{M}} |N(\theta,\phi)|R_j(\theta,\phi)\overline{R_k(\theta,\phi)}\exp\left(\frac{2\pi i}{\lambda}\left[d_j(\theta,\phi)-d_k(\theta,\phi)\right]\right)\right\} \qquad (20)$$

$$s_{kj} = \frac{1}{|\mathcal{F}_p \cap (\mathcal{M}\cup\mathcal{S})|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap(\mathcal{M}\cup\mathcal{S})} |N(\theta,\phi)|R_j(\theta,\phi)\overline{R_k(\theta,\phi)}\exp\left(\frac{2\pi i}{\lambda}\left[d_j(\theta,\phi)-d_k(\theta,\phi)\right]\right)\right\} . \qquad (21)$$

By Theorem 1, maximizing DIBCF_p requires the computation of any eigenvector corresponding to the largest eigenvalue for the eigenproblem (19). The numerical solution of (19) is fully discussed in the next section.

There are two considerations that should enter into the particular choice of p for the Fuller points $\mathcal{F}_p$. First, the points in $\mathcal{F}_p$ should be numerous enough to adequately sample the worst behavior of any realizable field pattern. In other words, p should be large enough that even the narrowest sidelobe achievable in the field pattern will contain points in $\mathcal{F}_p$. Second, Theorem 1 requires that the denominator matrix S of the DIBCF_p ratio be positive definite. Normally, the sampling criterion will effect this.

NUMERICAL SOLUTION OF THE EIGENPROBLEM: DIRECT METHOD

The eigenproblem (19) is equivalent to the eigenproblem

$$(S^{-1}\ M)z = \mu z \quad , \qquad (22)$$

In other words, the eigenvalues and eigenvectors of equation (22) are precisely the same as those of equation (19). There are two difficulties in using equation (22) for numerical computation. First, it requires the inverse of the matrix S, whose only special structure is that it is positive definite and Hermitian. In general, numerical computation of the inverse of matrices should be avoided if possible. Second, equation (22) is not a Hermitian eigenproblem; that is, $S^{-1}M$ is not necessarily Hermitian even though S and M are both Hermitian. This means that the eigenvalues and eigenvectors of equation (22) must be computed by a routine designed for a general complex matrix, and this means that the eigenvalues can (and do) turn out to be complex numbers because of numerical roundoff. Since Theorem 1 requires that all the eigenvalues be strictly real numbers, there is numerical error in using equation (22) caused by destruction of the natural Hermitian symmetry in equation (19). For these reasons, it is desirable to solve the eigenproblem (19) directly.

Martin and Wilkinson (reference 3) contain a method and a routine for solving this eigenproblem when M and S are real symmetric. Both the technique and the routine can be adapted to the Hermitian case. Every Hermitian positive definite matrix S has the Cholesky decomposition

$$S = L\,\bar{L}^T,$$

where L is a lower triangular matrix. Thus

$$Mz = \mu L\bar{L}^T z$$

$$L^{-1}\,Mz = \mu\bar{L}^T z$$

$$L^{-1}M(\bar{L}^{-T}\,\bar{L}^T)z = \mu\bar{L}^T z$$

$$(L^{-1}\,M\,\bar{L}^{-T})x = \mu x\ , \qquad (23)$$

where $x = \bar{L}^T z$. (24)

Therefore, the eigenvalues of $L^{-1}\,M\,\bar{L}^{-T}$ are precisely the eigenvalues of equation (19), and the eigenvectors x of $L^{-1}\,M\,\bar{L}^{-T}$ and the eigenvectors z of $S^{-1}M$ are related by equation (24). Note also that equation (23) is a Hermitian eigenproblem, since $L^{-1}\,M\,\bar{L}^{-T}$ is a Hermitian matrix. It is therefore possible to solve equation (23) by numerical methods designed for Hermitian eigenproblems which explicitly use the fact that the eigenvalues are real (reference 2). Therefore, the eigenvalues computed by using equation (23) will always be real as required.

This computational procedure seems to require a prohibitively large number of arithmetic operations; however, reference 2 shows that the computations may be done very efficiently because of the special structure of the matrices involved. For example, the matrix $L^{-1}\,M\bar{L}^{-T}$ can be computed (without inverting the matrix L) by using only 2/3 n^3 complex multiplications. This compares to 1/2 n^3 complex multiplications in the computation of S^{-1} alone in equation (23). In terms of storage required, computation time, and numerical accuracy, the use of equations (23) and (24) is preferable to equation (22).

The routine in reference 3 was adapted to the Hermitian case, using the routines available in reference 2 to solve the eigenproblem (23). This routine is called PENCLH, and its listing is available in appendix B. The listings of the routines used from reference 2 are not available because they are proprietary information under terms of the lease arrangements made with International Mathematical and Statistical Libraries, Incorporated. Finally, it is pointed out that the routine PENCLH computes all the eigenvalues and eigenvectors of equation (19), and not merely the largest eigenvalue and corresponding eigenvector(s).

NUMERICAL SOLUTION OF THE EIGENPROBLEM: INDIRECT METHOD

As discussed in the subsection COMPUTER STORAGE PROBLEM, the drawback to the direct method is excessive computer storage for large arrays. The group coordinate relaxation (or indirect) method overcomes this drawback, but at the cost of computer time and the loss of ability to compute the lower order eigenvalues/eigenvectors. The group coordinate relaxation method is detailed in reference 5 for the real symmetric eigenproblem $Ax = \mu x$. This method can be extended easily to the Hermitian eigenproblem

$$Mz = \mu Sz \ . \tag{25}$$

Although the method can be extended to arbitrary Hermitian matrices M and S, with S positive definite, it is important here to retain the structure of M and S as given by equations (20) and (21). The reason is that the Hermitian forms of M and S can be evaluated directly without knowledge of any of the entries in the matrices M and S. This is the fact that allows the computer storage problem to be overcome.

The following notation will be very useful. Define the basis vectors

$$e_1 = \langle 1\ 0\ 0 \ldots 0\ 0\rangle^T \tag{26a}$$

$$e_2 = \langle 0\ 1\ 0 \ldots 0\ 0\rangle^T$$

$$\vdots$$

$$e_n = \langle 0\ 0\ 0 \ldots 0\ 1\rangle^T .$$

Note that each of these vectors is of length n. To define vectors e_m for $m \geq n + 1$, we first set

$$t = \begin{cases} n & \text{, if m is an integer multiple of n} \\ m - [m/n]n, & \text{if not} \end{cases} \tag{26b}$$

where [ ] denotes the greatest integer function. Since equation (26b) requires $1 \leq t \leq n$, we can now define

$$e_m = e_t \ , \quad m \geq n + 1 \ . \tag{26c}$$

In other words, we have defined

$$\begin{aligned} e_1 &= e_{n+1} = e_{2n+1} = \dots \\ e_2 &= e_{n+2} = e_{2n+2} = \dots \\ &\vdots \\ e_n &= e_{2n} = e_{3n} = \dots \ . \end{aligned}$$

Before the group coordinate relaxation algorithm can begin, two items must be specified. First, an initial guess

$$a_{(0)} = \langle a_1^{(0)}, a_2^{(0)}, \dots, a_n^{(0)} \rangle^T \tag{27}$$

at the optimal element excitation vector is required. The vector $a_{(0)}$ cannot consist of only zero entries, but it is otherwise completely arbitrary. Second, it is decided in some manner to work with subarrays of the full array of size $r \geq 1$. It will be shown below that choosing to work with subarrays of size r will mean that generalized eigenproblems of size r+1 will have to be solved, so computer storage plays an important role in the choice of r. Another important consideration is computation time. In general, the larger r is taken to be, the faster optimum excitations of the full array can be computed.

The group coordinate relaxation algorithm is most easily described by exhibiting the first two steps of the algrithm. From these steps it is easy to see the general procedure. In the first step, we seek to

$$\underset{x \varepsilon Q_0}{\text{maximize}} \ \frac{\bar{x}^T M x}{\bar{x}^T S x} \ , \tag{28}$$

where Q_0 is the vector space of dimension r+1 whose general element, x, can be written in the form

$$x = c_0 \ a_{(0)} + c_1 \ e_1 + \dots + c_r \ e_r \tag{29}$$

for some complex constants $c_0, c_1, \dots, c_r$. It is shown below that (28) is a ratio of Hermitian forms in the parameters $\{c_0, c_1, \dots, c_r\}$. Therefore, by Theorem 1, the solution of (28) requires solving an eigenproblem of size r+1. Let

$$a_{(1)} = \tilde{c}_0 \ a_{(0)} + \tilde{c}_1 \ e_1 + \dots + \tilde{c}_r \ e_r \tag{30}$$

be a vector for which the maximum (28) is attained. This completes the first step. In the second step, we seek to

$$\underset{x \varepsilon Q_1}{\text{maximize}} \frac{\bar{x}^T M x}{\bar{x}^T S x}, \tag{31}$$

where Q_1 is the vector space of dimension r+1 whose general element, x, can be written in the form

$$x = c_0 \, a_{(1)} + c_1 \, e_{r+1} + \dots + c_r \, e_{2r} \tag{32}$$

for some complex constants $c_0, c_1, \dots, c_r$. Since (31) is again a ratio of Hermitian forms in the parameters $\{c_0, c_1, \dots, c_r\}$, we solve an eigenproblem of size r+1 to compute a vector

$$a_{(2)} = \overset{\approx}{c}_0 \, a_{(1)} + \overset{\approx}{c}_1 \, e_{r+1} + \dots + \overset{\approx}{c}_r \, e_{2r} \tag{33}$$

for which the maximum (31) is attained. This completes the second step. Continuing in this fashion defines the group coordinate relaxation algorithm.

We see that this algorithm "cycles" through the entire array using subarrays of size r. This is because the basis vectors $\{e_k\}$ are defined to "cycle" regularly through the vectors $\{e_1, e_2, \dots, e_n\}$. Also, if r does not divide n evenly, then each individual element belongs to a number of different subarrays as the computation proceeds. In other words, if r does not divide n, the entire array is not subdivided into disjoint subarrays.

The group coordinate relaxation algorithm generates a sequence of vectors $a_{(0)}, a_{(1)}, a_{(2)}, \dots$ which converges to the eigenvector corresponding to the largest eigenvalue of equation (19). Convergence is assured regardless of the starting vector, with some highly unlikely exceptions. These exceptions are easy to state. If any of the computed vectors $\{a_{(0)}, a_{(1)}, a_{(2)}, \dots\}$ is precisely an eigenvector of equation (19) which corresponds to an eigenvalue that is not the largest eigenvalue of the equation, then the group coordinate relaxation method will not move from this eigenvector. Numerical roundoff error will probably prevent this in practice. For further discussion and for a convergence theorem whose proof can be extended to the present situation, see reference 5. For an application of these mathematical methods to other problems, see reference 8.

An important feature is that the last computed vector $a_{(k)}$ gives a larger DIBCF_p than the previous vector $a_{(k-1)}$. This is easy to see by observing the ratios (28) and (31).

Another very useful observation is that the algorithm requires knowledge of only $a_{(k)}$ to compute $a_{(k+1)}$. This means that if computation must be interrupted for any reason, it is necessary to store only the last computed vector in order to restart computations.

We conclude this section by an examination of the maximum (28). Everything that is said of (28) is easily translated to the maximum (31), as well as all the other maxima required in the group coordinate relaxation algorithm. Note, first, that putting (29) into (28) gives the identity

$$\max_{x \varepsilon Q_0} \frac{\bar{x}^T M x}{\bar{x}^T S x} = \max_{z} \frac{\bar{z}^T G z}{\bar{z}^T B z} \quad , \tag{34}$$

where $z = \langle c_0, c_1, \ldots, c_r \rangle^T$. $G = [g_{ij}]$ and $B = [b_{ij}]$ are $(r+1) \times (r+1)$ Hermitian matrices whose general entries are given by

$$g_{00} = \bar{a}^T_{(0)} M a_{(0)}$$

$$g_{0j} = \bar{a}^T_{(0)} M e_j \quad , \quad j = 1, \ldots, r$$

$$g_{k0} = \bar{e}^T_k M a_{(0)} \quad , \quad k = 1, \ldots, r$$

$$g_{kj} = \bar{e}^T_k M e_j \quad , \quad k, j = 1, \ldots, r$$

and

$$b_{00} = \bar{a}^T_{(0)} S \, a_{(0)}$$

$$b_{0j} = \bar{a}^T_{(0)} S \, e_j \quad , \quad j = 1, \ldots, r$$

$$b_{k0} = \bar{e}^T_k S \, a_{(0)} \quad , \quad k = 1, \ldots, r$$

$$b_{kj} = \bar{e}^T_k S e_j \quad , \quad k, j = 1, \ldots, r \, .$$

Thus the entries of G and B are computable from the Hermitian forms of M and S, respectively. Let $V_0(\theta,\phi)$ be the field pattern of the entire array for the excitations $a_{(0)}$. Then we have, explicitly,

$$g_{00} = \frac{1}{|\mathcal{J}_p \cap \mathcal{M}|} \left\{ \sum_{(\theta,\phi) \varepsilon \mathcal{J}_p \cap \mathcal{M}} |N(\theta,\phi) V_0^2(\theta,\phi)| \right\} \tag{35}$$

$$g_{k0} = \frac{1}{|\mathcal{F}_p \cap \mathcal{M}|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap\mathcal{M}} |N(\theta,\phi)|\; V_0(\theta,\phi)\overline{R_k(\theta,\phi)}\; \exp\left[-\frac{2\pi i}{\lambda}\, d_k(\theta,\phi)\right]\right\}, \quad k = 1, \ldots, r \tag{36}$$

$$g_{0k} = \overline{g_{k0}}\,, \qquad k = 1, \ldots, r \tag{37}$$

$$g_{kj} = m_{kj}\,, \qquad k, j = 1, \ldots, r \tag{38}$$

where m_{kj} is given by (20), and similarly,

$$b_{00} = \frac{1}{|\mathcal{F}_p \cap (\mathcal{M}\cup\mathcal{S})|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap(\mathcal{M}\cup\mathcal{S})} |N(\theta,\phi)\; V_0^2(\theta,\phi)|\right\} \tag{39}$$

$$b_{k0} = \frac{1}{|\mathcal{F}_p \cap (\mathcal{M}\cup\mathcal{S})|}\left\{\sum_{(\theta,\phi)\in\mathcal{F}_p\cap(\mathcal{M}\cup\mathcal{S})} |N(\theta,\phi)|\, V_0(\theta,\phi)\overline{R_k(\theta,\phi)}\exp\left[-\frac{2\pi i d}{\lambda}{}_k(\theta,\phi)\right]\right\}, \quad k=1,\ldots,r \tag{40}$$

$$b_{0k} = \overline{b_{k0}}\,, \qquad k = 1, \ldots, r\,, \tag{41}$$

$$b_{kj} = s_{kj}\,, \quad k, j = 1, \ldots, r\,. \tag{42}$$

where s_{kj} is given by (21). Because $V_0(\theta,\phi)$ can be computed easily for each (θ,ϕ), we see that equations (35) through (42) can be computed efficiently in terms of time and core storage requirements. Now, by using Theorem 1, we see that the maximum of

$$\frac{\bar{z}^T G z}{\bar{z}^T B z}$$

is achieved by any vector

$$\tilde{z} = <\tilde{z}_0, \tilde{z}_1, \ldots, \tilde{z}_r>^T$$

which is an eigenvector of the largest eigenvalue of $Gz = \mu Bz$. Thus, from (34), we see that

$$a_{(1)} = \tilde{z}_0\, a_{(0)} + \tilde{z}_1\, e_1 + \ldots + \tilde{z}_r\, e_r$$

is a vector for which the maximum (28) is attained.

EXAMPLES

EXAMPLE 1: A COMPARISON WITH DOLPH-CHEBYSHEV DESIGN

This example serves two purposes. First, it provides a comparison with the Dolph-Chebyshev line array design. Second, it gives some insight into the nature of the lower order eigenvalues/eigenvectors.

Suppose that we have a line array of 15 elements which lies along the y-axis (see figure 1) with equal spacings of 0.5 wavelength, where the wavelength $\lambda=1$. (The units of length are irrelevant.) Thus, if the first element lies at the origin with coordinates (0., 0., 0.), the 15-th element has the coordinates (0., 7., 0.). It is well known that any line array has a field pattern with cylindrical symmetry about the array axis. Therefore, the most effective choices for $\mathcal{M}$ and $\mathcal{S}$ would seem to be cylindrically symmetric. We define $\mathcal{M}$ to be the set of all directions that lie within 8 degrees of a normal to the y-axis, and we define $\mathcal{S}$ to be the collection of all other directions. Hence, $\mathcal{M}$ is 16 degrees "wide." The ambient noise field is assumed to be flat, and the individual elements are assumed to be omnidirectional. Finally, considering the construction of the Fuller points $\mathcal{F}_p$, we choose p = 24.

The above data completely define the DIBCF$_p$ array problem. The rest is understanding the computer program usage (see appendix A). In appendix B, a listing of the entire program required for this example is given. The results of the execution are given in table 5. Also, the field pattern (in the xy-plane) is given in figure 4.

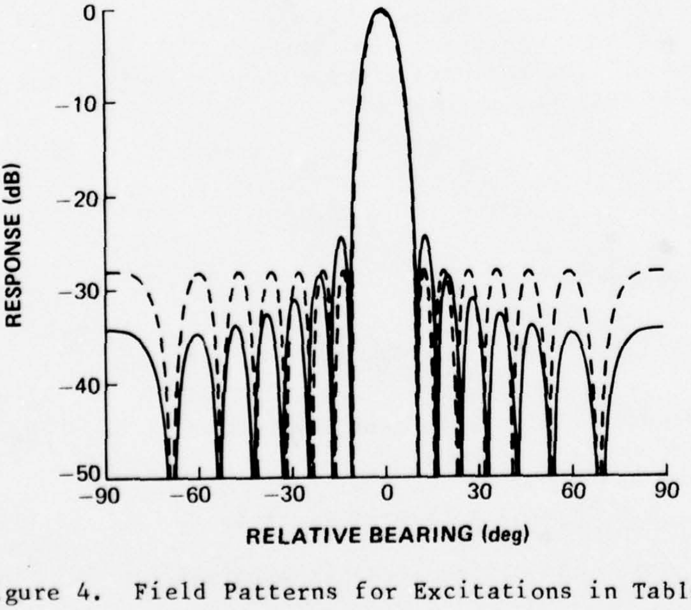

Figure 4. Field Patterns for Excitations in Table 5.

The Dolph-Chebyshev excitations are designed exclusively for half-wavelength equispaced line arrays with omnidirectional elements. For a given number of elements, the Dolph-Chebyshev excitations depend only upon the steered direction and on the specified sidelobe level. For a broadside (i.e., steered normal to the line of the array) 15-element array, the Dolph-Chebyshev excitations for a 28 dB sidelobe-level field pattern are given in table 5. The corresponding field pattern is shown in figure 4.

We note that the mainlobe shape of the Dolph-Chebyshev array and the DIBCF_p array are indistinguishable in figure 4. The only difference lies in sidelobe structure. We can also see that by sacrificing approximately 3 dB in the sidelobe nearest the mainlobe, all the remaining sidelobes can be made smaller than the overall 28 dB sidelobe level of the Dolph-Chebyshev array.

Table 5. Excitations for 15 Element Equispaced Line Array: Dolph-Chebyshev Vs DIBCF_{24}

Element No.	Dolph-Chebyshev	DIBCF_{24}
1	.34371	.25687
2	.35775	.39520
3	.50403	.54024
4	.65338	.68290
5	.79108	.81242
6	.90242	.91188
7	.97487	.97920
8	1.00000	1.00000
9	.97487	.97920
10	.90242	.91188
11	.79108	.81242
12	.65338	.68290
13	.50403	.54024
14	.35775	.39520
15	.34371	.25687

What about the lower order eigenvalues? The first four eigenvalues/eigenvectors are listed in table 6. (Note that the eigenvector of μ_1 in table 6 is also given in table 5, but is normalized differently.) Also, the corresponding field patterns are given in figures 5 through 8. We remark only that the field pattern for the largest eigenvalue μ_1 has no nulls in the mainlobe region $\mathcal{M}$, whereas the field pattern for μ_2 has one null in $\mathcal{M}$, two nulls for μ_3, and 3 nulls for μ_4.

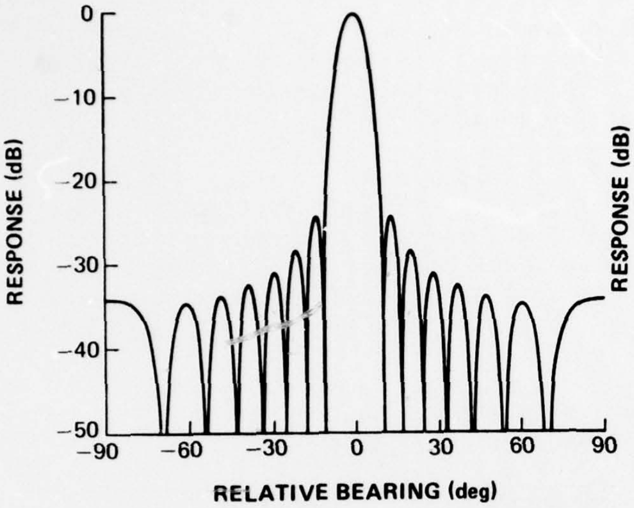

Figure 5.

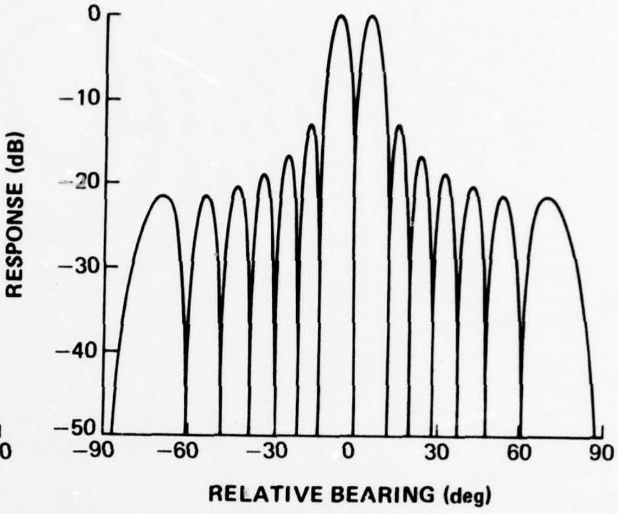

Figure 6.

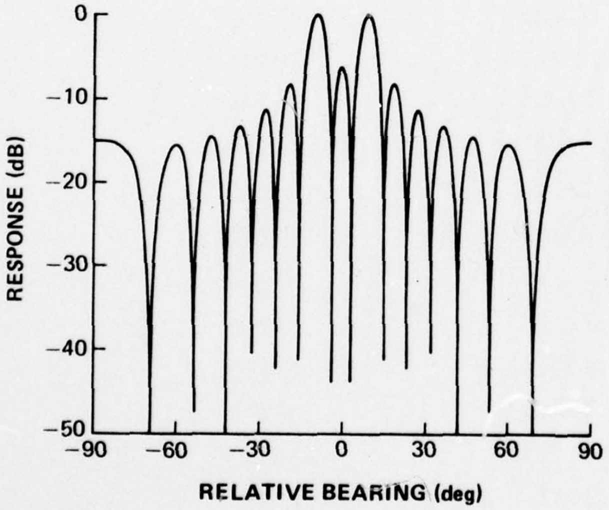

Figure 7.

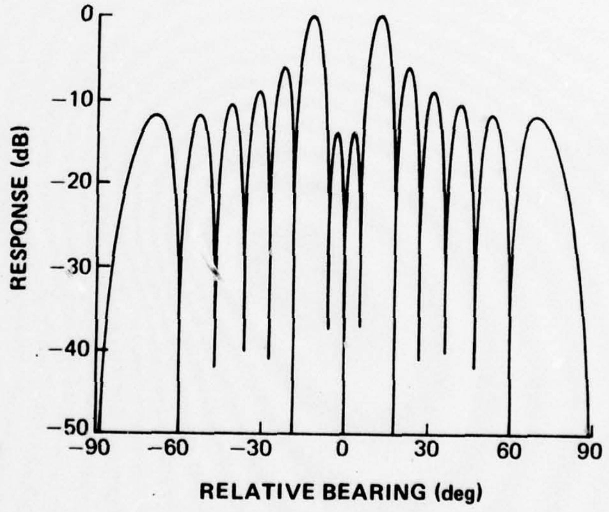

Figure 8.

Figures 5 through 8 Correspond to Eigenvectors One through Four, Respectively, of Example 1

Table 6. The Four Largest Eigenvalues/Eigenvectors of Example 1

Element No.	μ_1 = .9894	μ_2 = .8206	μ_3 = .3231	μ_4 = .0383
1	.0916	.2755	.4486	.5106
2	.1409	.3158	.3666	.2141
3	.1927	.3289	.2442	-.0356
4	.2436	.3112	.1066	-.2042
5	.2898	.2675	-.0242	-.2664
6	.3252	.1929	-.1413	-.2454
7	.3492	.1030	-.2097	-.1391
8	.3567	.0000	-.2403	.0000
9	.3492	-.1030	-.2097	.1391
10	.3252	-.1929	-.1413	.2454
11	.2898	-.2675	-.0242	.2664
12	.2436	-.3112	.1066	.2042
13	.1927	-.3289	.2442	.0356
14	.1409	-.3158	.3666	-.2141
15	.0916	-.2755	.4486	-.5106

EXAMPLE 2: A 105-ELEMENT CYLINDRICAL ARRAY

This example illustrates the use of subarrays (i.e., the group coordinate relaxation method) for computing optimum DIBC with limited computer storage. We select an array with 105 elements arranged around a cylinder. Specifically, we first construct 7 rings of 15 elements each, and then place the axis of each of these rings along the x-axis (see figure 9). The exact positions (and element numbers) are given in table 7, where the units of length are such that the wavelength $\lambda = 1$.

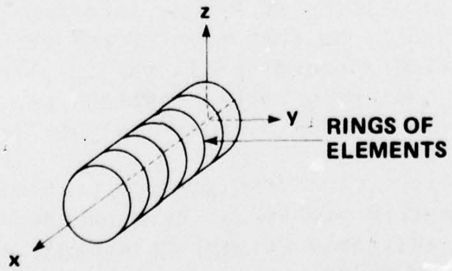

Figure 9. Arrangement of Elements in Example 2.

Each element of this array has a hemispherical field pattern defined in the following manner. We conceive of the array as being supported by a (transparent) cylinder. Through each element, we pass a tangent plane parallel to

Table 7. Coordinates of Elements in Example 2

Element	Coordinates		
No.	x	y	z
1	.0000	.7642	.0000
2	.3337	.7642	.0000
3	.6674	.7642	.0000
4	1.0011	.7642	.0000
5	1.3348	.7642	.0000
6	1.6685	.7642	.0000
7	2.0022	.7642	.0000
8-14	As above	.6982	.3108
15-21	↓	.5114	.5679
22-28		.2362	.7268
29-35		-.0799	.7600
36-42		-.3821	.6618
43-49		-.6183	.4492
50-56		-.7475	.1589
57-63		-.7475	-.1589
64-70		-.6183	-.4492
71-77		-.3821	-.6618
78-84		-.0799	-.7600
85-91		.2362	-.7268
92-98		.5114	-.5679
99-105		.6982	-.3108

the cylinder axis. The field pattern of an element has unit response on the side of the plane which DOES NOT contain the cylinder and has zero response on the side that DOES contain the cylinder.

We assume that the ambient noise field is flat. Also, we choose p = 32 in the definition of the Fuller points $\mathcal{F}_p$.

The mainlobe region $\mathcal{M}$ is defined as a "half cone" lying above the positive x-axis. Specifically, consider the solid cone with axis lying along the positive x-axis, with its vertex at the origin, and with a vertex angle of 40 degrees. The xy-plane slices this cone into two equal parts, and the mainlobe region $\mathcal{M}$ is defined to be that part of the cone which lies above (i.e., points having positive z-coordinates) the xy-plane. The sidelobe region $\mathcal{S}$ is defined to be the set of all directions that are not in the mainlobe region $\mathcal{M}$. There is no ignored region $\mathcal{I}$ in this example.

With all the above choices, the DIBCF$_p$ array problem is completely specified. To solve the entire problem at once on the UNIVAC 1108 requires more core storage than is available without resorting to indirect addressing or some other scheme, so we use subarrays to maximize the DIBCF$_p$ of the entire array. It seems best to use as many elements as can be easily handled with the computer storage available, so in this case we choose 69 elements; that is, roughly 2/3 of the array. This turns out to require only about 45,000 words of computer storage. To effect this computation, using the program in appendix B, requires that: (1) we use the computer program as if we were going to solve the entire problem without group coordinate relaxation, and (2) we add only 2 additional cards, namely, NRELAX = 69+1 and EPSI = .001.

The group coordinate relaxation scheme required roughly 1650 seconds per iteration, and 5 iterations in all. Thus total computation time was roughly 2 1/4 hours. Table 8 gives the final (optimal) set of element excitations. Figure 10 gives the vertical field pattern, and figure 11 gives the horizontal field pattern, for these excitations. We point out that in the field patterns in these two figures have abrupt jumps because the individual element field patterns also have sharp jumps due to their assumed hemispherical field patterns. (These field patterns were computed by the field pattern program described in reference 9.)

Table 8. Optimum Excitations for Example 2

Element No.	Magnitude	Phase	Element No.	Magnitude	Phase
1	.01497	-2.04150	54	.06917	- .20862
2	.04456	1.55478	55	.04197	-3.12157
3	.07601	-1.27318	56	.01513	.26034
4	.09175	2.15264	57	.01676	-2.08531
5	.08206	- .72447	58	.05236	1.52329
6	.05216	2.68649	59	.09072	-1.29956
7	.02126	- .15355	60	.10997	2.13497
8	.01160	- .72850	61	.09776	- .74037
9	.03588	2.55205	62	.06142	2.67618
10	.06133	- .39698	63	.02515	- .16489
11	.0734	2.93476	64	.01874	-2.67888
12	.06403	.00079	65	.05833	.97472
13	.03893	-2.92334	66	.10512	-1.80774
14	.01398	.46931	67	.13227	1.65509
15	.00348	- .01323	68	.12180	-1.18257
16	.01278	3.13121	69	.07979	2.25832
17	.02296	.29855	70	.03212	- .58940
18	.02717	-2.50653	71	.03656	2.70718
19	.02319	.98764	72	.08537	- .05512
20	.01409	-1.73254	73	.12945	-2.85812
21	.00570	1.78955	74	.14265	.61788
22	.01032	1.68325	75	.11732	-2.17831
23	.02624	-1.20492	76	.06927	1.34010
24	.04616	2.09737	77	.02574	-1.30699
25	.05783	- .90429	78	.07023	2.82735
26	.05388	2.37697	79	.18862	.01720
27	.03679	- .61724	80	.30892	-2.83428
28	.01519	2.70356	81	.35755	.59043
29	.03297	1.32646	82	.30468	-2.26194
30	.08731	-1.59998	83	.18437	1.18512
31	.146261	1.74147	84	.06783	-1.59534
32	.17553	-1.20741	85	.05600	2.52014
33	.15743	2.12465	86	.13959	- .31104
34	.10196	- .81950	87	.21923	3.11525
35	.04099	2.52482	88	.24587	.25434
36	.00197	1.82093	89	.20337	-2.59581
37	.00307	-1.47902	90	.11910	.86311
38	.00857	1.58849	91	.04179	-1.86568
39	.01455	-1.39023	92	.02626	2.88925
40	.01616	1.95539	93	.06438	.22940
41	.01282	- .94279	94	.10232	-2.51417
42	.00653	2.49085	95	.11889	1.00683
43	.00811	.10499	96	.10393	-1.75955
44	.02391	-2.95328	97	.06623	1.76756
45	.04158	.42416	98	.02674	- .95527
46	.05011	-2.47388	99	.01431	-2.54878
47	.04422	.92678	100	.04566	1.15251
48	.02716	-1.92583	101	.08178	-1.63309
49	.01048	1.50198	102	.10300	1.82950
50	.01266	-1.16074	103	.09445	-1.01612
51	.03973	2.24429	104	.06124	2.42174
52	.06734	- .65477	105	.02521	- .41890
53	.07980	2.70791			

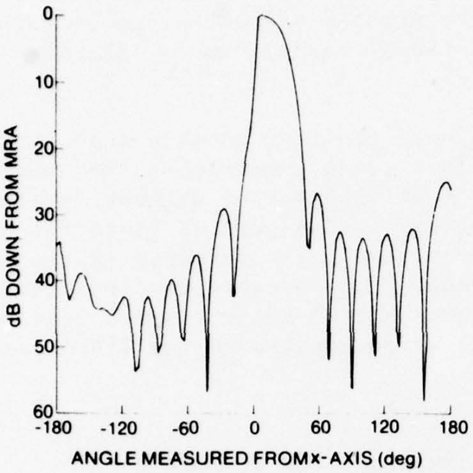

Figure 10. Vertical Beam Pattern for Example 2 with Excitations Given in Table 8.

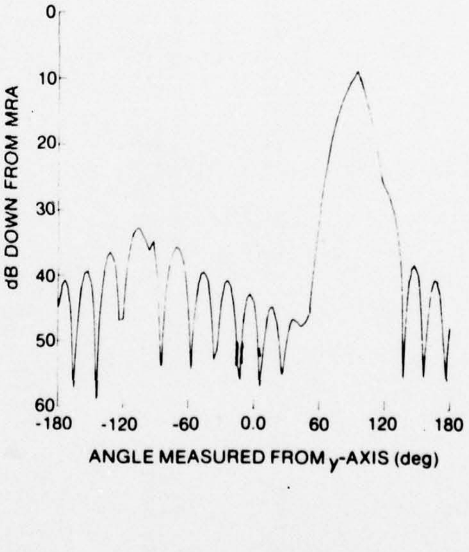

Figure 11. Horizontal Beam Pattern for Example 2 with Excitations Given in Table 8.

This method creates a steadily increasing sequence of estimates for the largest eigenvalue. Since there were 5 iterations, there were 5 estimates, and these are given in table 9. Based on the plots in figures 10 and 11, it would seem that the field pattern cannot be improved significantly. In other words, even if we had chosen the variable EPSI to be even smaller, we would have forced additional (expensive) iterations of the algorithm which would probably not have improved the field patterns.

Table 9. Group Coordinate Relaxation Estimates of Largest Eigenvalue for Example 2

Iteration No.	Estimate of Largest Eigenvalue
1	.91427
2	.96100
3	.96374
4	.96532
5	.96559

EXAMPLE 3: EFFECTS OF SAMPLING

The first two examples did not mention the effects of sampling on the field patterns. Specifically, the parameter p in the definition of the Fuller points $\mathfrak{F}_p$ determines how finely we have sampled all spatial directions. Hence, the parameter p has a strong influence on the resulting field patterns. In particular, if p is not sufficiently large, then it is possible for the optimal (DIBCF$_p$) field pattern to have a split beam.

We illustrate this effect by systematically varying p in the array of example 1, but for a different choice of $\mathcal{M}$ and $\mathcal{S}$. Here, we define $\mathcal{M}$ to be the collection of all directions whose projection onto the xz-plane lies within ±8 degrees of the z-axis. Specifically, the direction corresponding to direction cosines (α,β,γ) lies in $\mathcal{M}$ only if $|\alpha/\sqrt{\alpha^2 + \gamma^2}| \le \sin 8^o$. In other words, $\mathcal{M}$ consists of all directions contained "between" the two planes intersecting the yz-plane at the angles of +8 degrees and -8 degrees. The sidelobe region $\mathcal{S}$ consists of all remaining directions, so there is no ignored region $\mathcal{I}$. Optimal excitations for several choices of p are given in table 10. The field patterns for p = 24 and p = 16 are given in figure 12. We do not present the field patterns for p = 32 and p = 40, because they are so similar to p = 24.

Table 10. Effects of Sampling on Excitations for Example 3

Element		p		
No.	16	24	32	40
1	.2953	.1275	.1322	.1177
2	-.0064	.1676	.1792	.1662
3	.3846	.2207	.2406	.2232
4	-.1045	.2497	.2613	.2581
5	.2990	.2884	.2946	.2934
6	-.2830	.3067	.2991	.3099
7	.1753	.3337	.3154	.3259
8	-.3280	.3346	.3117	.3279
9	.1753	.3337	.3154	.3259
10	-.2830	.3067	.2991	.3099
11	.2990	.2884	.2946	.2934
12	-.1045	.2497	.2613	.2581
13	.3846	.2207	.2406	.2232
14	-.0064	.1676	.1792	.1662
15	.2953	.1275	.1322	.1177
Largest eigenvalue, μ	.1149	.1243	.1165	.1225
No. of Fuller points, $\mathfrak{F}_p$	2562	5762	10242	16002

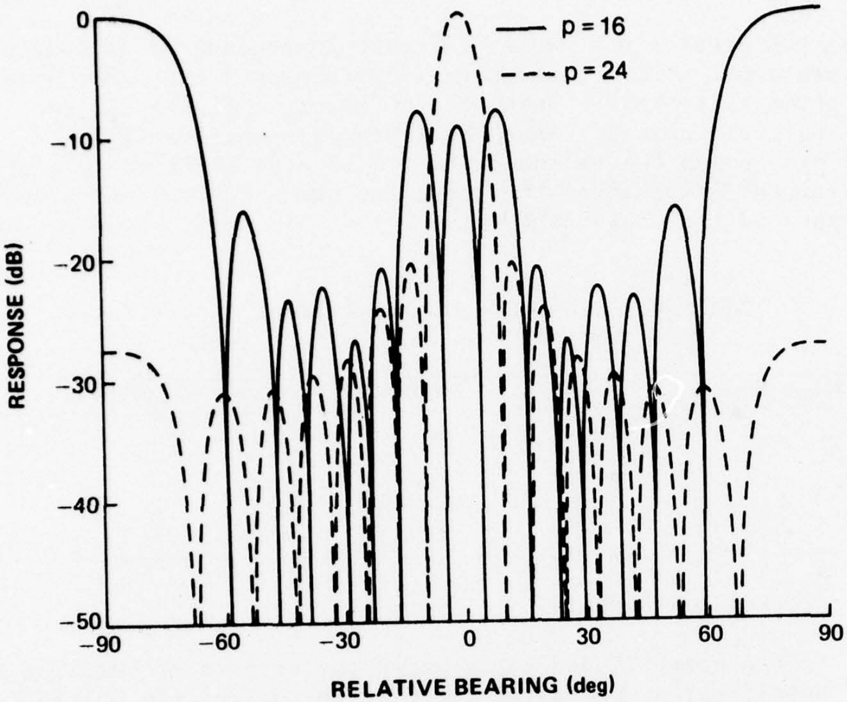

Figure 12. Field Pattern for Example 3 for p = 16.

Table 10 also shows the effects of oversampling. Note that the optimal excitations for p ≥ 24 are all similar, but they do not seem to be converging to an optimal set. This is probably due to the buildup of numerical round-off error in the required sums (i.e., equations (20) and (21)), but it could also be that p must be chosen even larger than 40 before the optimal excitations give the appearance of convergence. In any event, the importance of sampling sufficiently finely is clear, but evidently oversampling wastes time and increases the numerical round-off error in the computed optimal excitations.

EXAMPLE 4: TIME AND ACCURACY IN THE INDIRECT METHOD

It is clear from the definition of the indirect, or group coordinate relaxation method, that the size of the subarrays used and the stopping criteria for the iteration procedure both have significant effects on numerical accuracy of the computed excitations and on the time required to compute them. This example illustrates how numerical accuracy and computation time depend on both these parameters.

We consider a line array of 25 elements that lies along the y-axis with equal spacings of 0.5 wavelength, where the wavelength λ = 1. Thus the coordinates of the first and last elements are (0., 0., 0.) and (0., 12., 0.) respectively. We select the mainlobe region $\mathcal{M}$ to be the set of all directions that lie within 5 degrees of a normal to the y-axis, and we define $\mathcal{S}$

to be the set of all other directions. There is no ignored region $\mathcal{A}$. The ambient noise field is flat and the individual elements are assumed omnidirectional. Finally, we select the Fuller points $\mathcal{F}_{28}$. All the above completely define our problem.

Table 11 shows the number of iterations required for various choices of NRELAX (i.e., subarray size) and EPSI (i.e., stopping criteria). See appendix A. As can be expected, the number of iterations required increases with decreasing EPSI and decreases with increasing NRELAX. Also, the computation time per iteration increases with NRELAX.

Table 11. Number of Iterations Required in Example 4

	EPSI			
NRELAX-1	10^{-3}	10^{-4}	10^{-5}	Time/Iteration (sec)
5	9	13	24	83
10	5	10	13	95
15	3	4	6	115
20	3	3	3	140
25	1	1	1	172

An important concern is the numerical accuracy of the computed excitations. This is particularly important in light of the fact that numerical computation of eigenvectors by any method is considerably less stable than the numerical computation of eigenvalues. Table 12 shows the results obtained for NRELAX-1 = 5 by stopping after the first four complete passes through the array, i.e., for iterations 5, 10, 15, and 20, respectively. The exact results are included also. The field patterns corresponding to excitations of iteration 5 and the exact excitations are shown in figure 13. Note that at the end of iteration 5 the field pattern already possesses sidelobes in the correct positions, although they are about 3 dB higher than in the field pattern of the exact excitations. Thus, the effect of later iterations is to beat down the sidelobes while maintaining the mainlobe beamwidth.

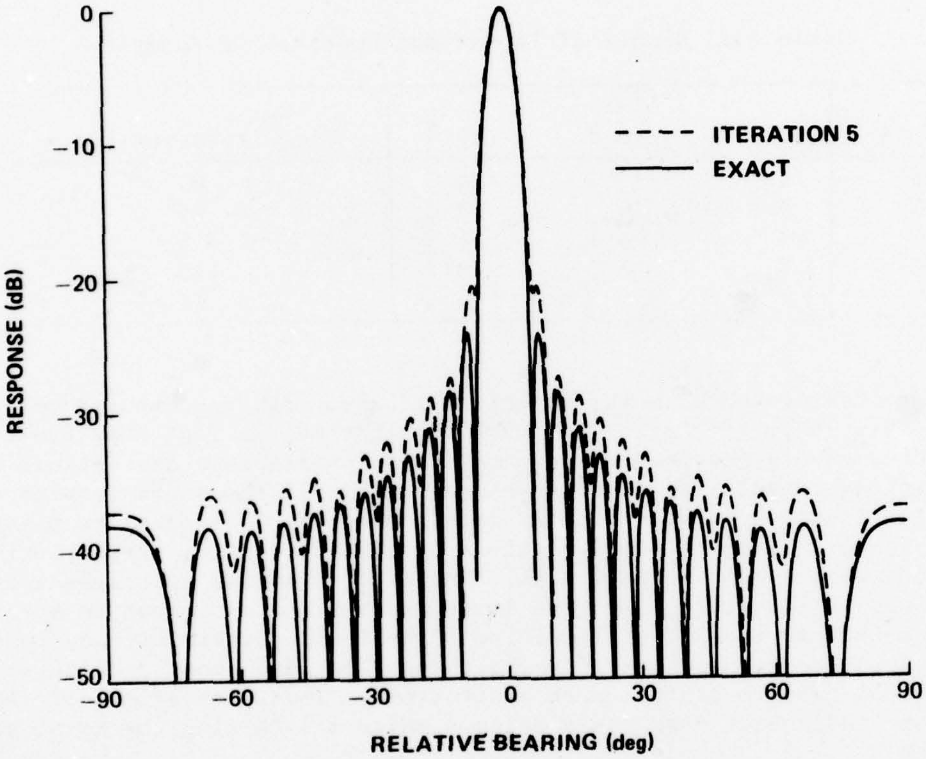

Figure 13. Comparison in Example 4: Exact Vs Iteration 5

Table 12. Example 4 with Subarrays of 5 Elements

Element No.	Iteration 5	Iteration 10	Iteration 15	Iteration 20	Exact
1	.632	.410	.623	.793	1.000
2	.813	.622	.887	1.090	1.339
3	.993	.860	1.173	1.406	1.692
4	1.183	1.117	1.428	1.736	2.057
5	1.394	1.398	1.800	2.082	2.436
6	1.446	1.788	2.218	2.461	2.792
7	1.648	2.079	2.539	2.795	3.145
8	1.840	2.355	2.832	3.095	3.456
9	2.023	2.611	3.100	3.366	3.732
10	2.188	2.842	3.328	3.592	3.955
11	2.541	3.143	3.522	3.783	4.120
12	2.664	3.298	3.654	3.904	4.223
13	2.749	3.392	3.722	3.956	4.254
14	2.806	3.439	3.737	3.951	4.223
15	2.809	3.421	3.677	3.877	4.120
16	3.011	3.271	3.559	3.729	3.955
17	2.951	3.145	3.396	3.541	3.732
18	2.834	2.967	3.174	3.298	3.456
19	2.696	2.756	2.929	3.022	3.145
20	2.489	2.494	2.631	2.700	2.792
21	2.012	2.190	2.290	2.353	2.436
22	1.762	1.887	1.957	2.000	2.057
23	1.511	1.567	1.632	1.658	1.692
24	1.260	1.294	1.313	1.324	1.339
25	1.000	1.000	1.000	1.000	1.000
μ_{max}	.9798313	.9857357	.9880019	.9886998	.9889890

SUMMARY

The concept of Directivity Index with Beamwidth Control (DIBC) has been defined as the ratio of power in the mainlobe region to the total power in both the mainlobe and the sidelobe regions. A mathematically and numerically tractable method for the computation of optimum element excitations (i.e., excitations which maximize DIBC) is presented. A technique known as group coordinate relaxation is shown to be an effective means of computing optimum element excitations for arrays of arbitrary numbers of elements, yet it requires only nominal core storage. Conceptually, the group coordinate relaxation technique employs subarrays of the full array in a systematic manner to optimize excitations of the full array. Four examples have been included, one of which demonstrates the effectiveness of group coordinate relaxation for a cylindrical array of 105 elements.

REFERENCES

1. F. R. Gantmacher, The Theory of Matrices, Chelsea Publishing Co., New York, NY, 1960.

2. International Mathematical and Statistical Libraries, The IMSL Library, Volume 2, 6th ed., IMSL, Inc., Houston, TX, 1977.

3. R. S. Martin and J. H. Wilkinson, "Reduction of the Symmetric Eigenproblem $Ax = \lambda Bx$ and Related Problems to the Standard Form," Numerische Mathematik, vol. 11, 1968, pp. 99-110.

4. D. Slepian and H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty--I," Bell System Technical Journal, vol. 40, 1961, pp.43-63.

5. D. K. Faddeev and V. N. Faddeeva, Computational Methods of Linear Algebra, W. H. Freeman & Co., San Francisco, CA, 1963, pp. 334-341.

6. T. W. Melnyk, O. Knop, and W. R. Smith, "External Arrangements of Points and Unit Charges on a Sphere: Equilibrium Configurations Revisited," Canadian Journal of Chemistry, vol. 55, 1977, pp. 1745-1761.

7. J. Prenis, "An Introduction to Domes," in The Dome Builder's Handbook, J. Prenis, ed., Running Press, Philadelphia, PA, 1973.

8. D. Lee, Maximization of Reverberation Index, NUSC Technical report 5375, Naval Underwater Systems Center, New London, CT, 2 October 1976.

9. D. Lee and G. A. Leibiger, Computation of Beam Patterns and Directivity Indices for Three-Dimensional Arrays with Arbitrary Element Spacings, NUSC Technical Report 4687, Naval Underwater Systems Center, New London, CT, 22 February 1974.

APPENDIX A

PROGRAM USAGE

The user is required to write between 2 and 4 routines which are to be used in conjunction with the elements in the program file STREIT*RELAX. The setup and use of each of these routines is easily seen from the listings in appendix B.

SUBROUTINE PHONE - Given a set of direction cosines (α, β, γ), compute the directional response of every individual element in the direction (α, β, γ). If all elements are omnidirectional, this subroutine need not be written by the user.

FUNCTION DNOISE - Given a set of direction cosines (α, β, γ), compute the magnitude of the ambient noise field in the direction (α, β, γ). If the ambient noise field is flat, this function subroutine need not be written by the user.

SUBROUTINE REGION - Given a set of direction cosines (α, β, γ), decide (by user prerogative) if this direction lies in the mainlobe region $\mathcal{M}$, the sidelobe region $\mathcal{S}$, or the ignored region $\mathcal{I}$. This subroutine <u>must</u> always be written by the user.

MAIN PROGRAM - This routine sets up the array problem and handles all the required computer allocations. Because of the computer storage problem, all the dimensional storage arrays (not to be confused with the antenna array) are defined by means of one PARAMETER statement. Therefore, the main routine must be recompiled for each different array. The following variables must be specified within the main routine:

N - Total number of elements in the antenna array. N MUST BE DEFINED BY MEANS OF A PARAMETER STATEMENT.

NRELAX - If group coordinate relaxation is not to be used, NRELAX = N. If group coordinate relaxation is to be used, NRELAX is 1 <u>plus</u> the number of elements to be relaxed. In either case, NRELAX MUST BE DEFINED BY MEANS OF A PARAMETER STATEMENT.

X(I)
Y(I)
Z(I) } - The coordinates of the I-th element, I = 1, 2, ..., N.

WAVLEN = Wavelength of the design frequency in units compatible with the element coordinates.

INOISE = { 0 if omnidirectional noise field
1 if noise field is specified by subroutine DNOISE.

IPHONE = { 0 if every element in the antenna array has an omnidirectional field pattern
1 if at least one element has a directional field pattern specified by subroutine PHONE.

INDEX = { 1 if ordinary directivity index is to be maximized.
0 if not.

ALPHAD, BETAD, GAMMAD - The (α, β, γ) direction cosines of the direction in which to maximize DI, provided INDEX = 1. If INDEX = 0, these variables are not needed.

ISTEER = { 0 if the array is unsteered
1 if the array is to be steered

ALPHAS, BETAS, GAMMAS - The (α, β, γ) direction cosines of the direction in which to steer the array, if ISTEER = 1. If ISTEER = 0, the default values are ALPHAS = BETAS = GAMMAS = 0.

NDIV - The parameter of the Fuller points $\mathcal{F}_p$; that is, NDIV = p.

RATIO - On return, an estimate of the mainlobe power to sidelobe power. If INDEX = 1, an estimate of DI. (This ratio can be unreliable.)

COEF - On input, COEF is unimportant if NRELAX = N. If NRELAX < N, then COEF contains the initial guess at all N of the final excitations. On return, COEF contains the optimal coefficients.

ICOEF = is the number of coefficient sets to print; i.e., print the ICOEF largest eigenvalue/eigenvector sets. $1 \le$ ICOEF $\le$ N. (Cannot be used for NRELAX < N.)

If NRELAX < N, then the following additional variables are needed:

LOOPMX = maximum number of relaxation iterations to perform.

EPSI = stopping criteria. Stop if

$$\left| 1 - \frac{\text{old eigenvalue estimate}}{\text{new eigenvalue estimate}} \right| \le \text{EPSI}.$$

IPRINT = { 0, if intermediate printout not desired
1, if intermediate printout is desired

IPUNCH = { 0, if no card output is desired.
1, if all newly computed excitations, real and imaginary parts, are to be punched in FORMAT (3(I4,2E11.6)).

The listing of the program in appendix B is also the listing of the code required for the particular DIBCF$_p$ array problem of example 1. The run stream is

```
@ RUN (user's RUN card)

@ HDG FIRST EXAMPLE

@ ASG,A STREIT*RELAX.

@ ASG,A IMSL*MATHLIB.

@ FOR,IS STREIT*RELAX.MAIN

@ FOR,IS STREIT*RELAX.REGION

@ FOR,IS STREIT*RELAX.PHONE  )
                              } NOT NEEDED IN EXAMPLE 1
@ FOR,IS STREIT*RELAX.DNOISE )

@ MAP,IS
  IN STREIT *RELAX.

  LIB IMSL *MATHLIB.
  END
@ XQT
```

APPENDIX B

PROGRAM LISTING

```
STREIT*RELAX(1).MAIN
C      SUBROUTINE MAIN   WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
C
C      THIS MAIN PROGRAM SETS UP AN ARRAY CONFIGURATION AND USES GROUP
C      COORDINATE RELAXATION TO SOLVE FOR OPTIMUM ELEMENT CURRENTS.
C
C       N = TOTAL NUMBER OF ELEMENTS IN THE ARRAY
C       NRELAX = 1 + THE NUMBER OF ELEMENT CURRENTS TO 'RELAX' IN EACH STEP
C       NDIV = PARAMETER CONTROLLING THE NUMBER OF POINTS CHOSEN ON THE SPHERE
C       X(I) = X COORDINATE OF I-TH ELEMENT
C       Y(I) = Y COORDINATE OF I-TH ELEMENT
C       Z(I) = Z COORDINATE OF I-TH ELEMENT
C       WAVLEN = WAVELENGTH OF DESIGN FREQUENCY
C       LOOPMX = MAX NUMBER OF RELAXATIONS TO PERFORM. ON RETURN, NUMBER DONE.
C       EPSI = STOPPING CRITERION...STOP IF IMPROVEMENT IS NOT AT LEAST EPSI
C       COEF = INITIAL GUESS AT ELEMENT CURRENTS. ON RETURN, FINAL ANSWER.
C       RATIO = ON RETURN, AN ESTIMATE OF MAINLOBE POWER TO SIDELOBE POWER
C       INOISE = 1 IF NOISE FIELD PRESENT,  = 0 IF NOT
C       IPHONE = 1 IF DIRECTIONAL ELEMENTS ARE USED,  = 0 IF NOT
C       INDEX = 1 IF DIRECTIVITY INDEX IS TO BE MAXIMIZED,  = 0 IF NOT
C       (ALPHAD,BETAD,GAMMAD)= DIRECTION IN WHICH TO MAXIMIZE  DI IF INDEX = 1
C       ISTEER = 1 IF ARRAY IS STEERED,  = 0 IF NOT
C       (ALPHAS,BETAS,GAMMAS) = DIRECTION COSINES OF ARRAY IF ISTEER = 1
C       IPRINT = 1 IF INTERMEDIATE PRINTOUT IS DESIRED,  = 0 IF NOT
C       IPUNCH = 1 IF INTERMEDIATE COEFFICIENTS ARE TO BE PUNCHED,  = 0 IF NOT
C       ICOEF = NUMBER OF COEF SETS TO PRINT (ONLY FOR CASE N=NRELAX )
C       THE REMAINDER OF THE ARRAYS ARE WORK AREAS
C
       PARAMETER N=15,NRELAX=15
C
       PARAMETER N3=3*NRELAX
       COMMON INOISE,IPHONE,WAVLEN,INDEX,ALPHAD,BETAD,GAMMAD,ICOEF,
     1        ISTEER,ALPHAS,BETAS,GAMMAS,NDIV,LOOPMX,EPSI,IPRINT,IPUNCH
       COMPLEX AMAIN(NRELAX,NRELAX),ASIDE(NRELAX,NRELAX),COEF(N),
     1        VX(NRELAX,NRELAX),DIRECT(N),DC(NRELAX)
       DIMENSION X(N),Y(N),Z(N),DL(N),WK(N3),D(NRELAX),INDX(NRELAX),
     1        V(NRELAX)
C
C      ************************************
C      BEGIN CODE FOR SPECIFIC ARRAY PROBLEM
C      ************************************
C
       DO 3 K=1,N
       X(K)=0.
       Y(K)=(K-1.)/2.
     3 Z(K)=0.
       WAVLEN=1.
       INDEX=0
       ISTEER=0
       INOISE=0
       IPHONE=0
       IPRINT=0
       IPUNCH=0
       EPSI=1.E-3
       ALPHAS=0.
       BETAS=0.
       GAMMAS=0.
       ALPHAD=ALPHAS
       BETAD=BETAS
       GAMMAD=GAMMAS
       ICOEF=15
       NDIV=24
C
C      ************************************
C      END CODE FOR SPECIFIC ARRAY PROBLEM
C      ************************************
C
       CALL RELAX(N,NRELAX,X,Y,Z,COEF,RATIO,AMAIN,ASIDE,DL,D,V,VX,WK,
     1        INDX,DIRECT,DC)
       END

@PRT,S R.RELAX
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
STREIT*RELAX(1).RELAX
      C      SUBROUTINE RELAX  WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
      C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
      C
      C      SUBROUTINE RELAX USES GROUP COORDINATE RELAXATION TO COMPUTE
      C      OPTIMUM ELEMENT CURRENTS .
      C
      C       N = NUMBER OF ELEMENTS IN THE ARRAY
      C       NRELAX = 1 + THE NUMBER OF ELEMENT CURRENTS TO 'RELAX'
      C       X(I) = X COORDINATE OF I-TH ELEMENT
      C       Y(I) = Y COORDINATE OF I-TH ELEMENT
      C       Z(I) = Z COORDINATE OF I-TH ELEMENT
      C       COEF = INITIAL GUESS AT ELEMENT CURRENTS
      C       RATIO = ON RETURN, AN ESTIMATE OF MAINLOBE POWER TO SIDELOBE POWER
      C       AMAIN = MATRIX OF QUADRATIC FORM FOR MAINLOBE REGION
      C       ASIDE = MATRIX OF QUADRATIC FORM FOR SIDELOBE REGION
      C       THE REMAINDER OF THE ARRAYS ARE WORK AREAS
      C
             SUBROUTINE RELAX(N,NRELAX,X,Y,Z,COEF,RATIO,AMAIN,ASIDE,DL,D,V,VX,
            1                  WK,INDX,DIRECT,DC)
      C
             DIMENSION X(1),Y(1),Z(1),DL(1),P(3,12),Q(3,12),D(1),INDX(1),WK(1),
            1          V(1),HTIME(2)
             COMPLEX AMAIN(NRELAX,1),ASIDE(NRELAX,1),VX(NRELAX,1),COEF(1),
            1        DIRECT(1),DC(1)
             COMMON INOISE,IPHONE,WAVLEN,INDEX,ALPHAD,BETAD,GAMMAD,ICOEF,
            1       ISTEER,ALPHAS,BETAS,GAMMAS,NDIV,LOOPMX,EPSI,IPRINT,IPUNCH
             COMMON/EXTRA/PI2LAM,ITINDX
             DATA ((P(I,J),I=1,3),J=1,12)/0.,0.,1.,
            1                                 .894427191,0.,.447213595,
            2                                 .276393202,.850650808,.447213595,
            3                                 .723606798,.525731112,-.447213595,
            4                                 .723606798,-.525731112,-.447213595,
            5                                 .0,.0,-1.,
            6                                 -.723606798,.525731112,.447213595,
            7                                 -.723606798,-.525731112,.447213595,
            8                                 .276393202,-.850650808,.447213595,
            9                                 -.276393202,.850650808,-.447213595,
            1                                 -.894427191,.0,-.447213595,
            2                                 -.276393202,-.850650808,-.447213595/
             PRINT 220
         220 FORMAT(1H1,' THE ELEMENT POSITIONS IN CARTESIAN COORDINATES ARE ',
            1/,16X,'ELT. NO.', 7X,'X',15X,'Y',15X,'Z')
             PRINT 219,(K,X(K),Y(K),Z(K),K=1,N)
         219 FORMAT(120,3F16.8)
             PRINT 225,WAVLEN
         225 FORMAT(/,' WAVELENGTH OF DESIGN FREQUENCY = ',E14.7)
             IF(INOISE.EQ.1)GO TO 201
             PRINT 202
         202 FORMAT(/,' NO AMBIENT NOISE FIELD PRESENT.')
             GO TO 212
         201 PRINT 203
         203 FORMAT(/,' AMBIENT NOISE FIELD IS CONSIDERED.')
         212 IF(IPHONE.EQ.1)GO TO 205
             PRINT 204
         204 FORMAT(/,' OMNIDIRECTIONAL ELEMENTS ARE USED.')
             GO TO 206
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
  205 PRINT 207
  207 FORMAT(/,' DIRECTIONAL ELEMENTS ARE USED.')
  206 IF(INDEX.EQ.1)GO TO 210
      PRINT 208
  208 FORMAT(/,' GENERALIZED DIRECTIVITY INDEX IS MAXIMIZED.')
      GO TO 209
  210 PRINT 211,ALPHAD,BETAD,GAMMAD
  211 FORMAT(/,' DIRECTIVITY INDEX IS MAXIMIZED WITH RESPECT TO DIRECTIO
     1N COSINES (',E12.7,',',E12.7,',',E12.7,')')
  209 IF(ISTEER.EQ.1)GO TO 213
      PRINT 214
  214 FORMAT(/,' THE ARRAY IS UNSTEERED.')
      GO TO 215
  213 PRINT 216,ALPHAS,BETAS,GAMMAS
  216 FORMAT(/,' THE ARRAY IS STEERED WITH DIRECTION COSINES (',E12.7,
     1',',E12.7,',',E12.7,')')
  215 NPOINT=10*NDIV**2+2
      PRINT 217,NPOINT,NDIV
  217 FORMAT(/,' NUMBER OF POINTS CHOSEN IN SPACE = ',I10,'    (DIVISION
     1S OF EACH FACE OF ICOSAHEDRON =',I5,')')
      IF(NDIV.GE.1)GO TO 3
      PRINT 2,NDIV
    2 FORMAT(//' ***** ILLEGAL PARAMETER IN SUBROUTINE RELAX,  NDIV = ',
     1           I5,' IMMEDIATE RETURN TO MAIN PROGRAM.')
      RETURN
    3 IF(N.GE.NRELAX)GO TO 15
      PRINT 2021,N,NRELAX
 2021 FORMAT(///' NRELAX CANNOT BE LARGER THAN N.  N = ',I10,'NRELAX = '
     1,I10,' RETURN TO MAIN PROGRAM.')
      RETURN
   15 IF(ISTEER.NE.1)GO TO 16
      SINIT=ALPHAS*ALPHAS+BETAS*BETAS+GAMMAS*GAMMAS
      IF(SINIT.GE..99999.AND.SINIT.LE.1.00001)GO TO 16
      PRINT 17
   17 FORMAT(////' ARRAY IS STEERED BUT THE SUM OF SQUARES OF DIRECTION
     1COSINES IS NOT ONE.',/,' RETURN TO MAIN PROGRAM.')
      RETURN
   16 IF(INDEX.NE.1)GO TO 18
      SINIT=ALPHAD*ALPHAD+BETAD*BETAD+GAMMAD*GAMMAD
      IF(SINIT.GE..99999.AND.SINIT.LE.1.00001)GO TO 18
      PRINT 19
   19 FORMAT(////' DIRECTIVITY INDEX IS TO BE MAXIMIZED BUT THE SUM OF S
     1QUARES OF DIRECTION COSINES IS NOT ONE.',/,' RETURN TO MAIN PROGRA
     2M.')
      RETURN
   18 IF(N.NE.NRELAX)GO TO 20
      IF(ICOEF.LE.0)ICOEF=1
      IF(ICOEF.GT.N)ICOEF=N
      GO TO 21
   20 ICOEF=1
   21 PI2LAM=2.*3.14159265  /WAVLEN
      VSAVE=-1.
      NRELX1=NRELAX-1
      SINIT=0.
      IF(ISTEER.NE.0)GO TO 4
      ALPHAS=0.
      BETAS=0.
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      GAMMAS=0.
    4 DO 1 I=1,N
      SINIT=SINIT+ABS(REAL(COEF(I))) + ABS(AIMAG(COEF(I)))
    1 DL(I)=X(I)*ALPHAS+Y(I)*BETAS+Z(I)*GAMMAS
      IF(SINIT.GE..001)GO TO 6
      DO 8 I=1,N
    8 COEF(I)=CMPLX(1.,0.)
    6 IF(LOOPMX.GE.1)GO TO 9
      LOOPMX= 25*((N/NRELX1)+1)
    9 IF(N.EQ.NRELAX)LOOPMX=1
      IF(IPRINT.EQ.0)GO TO 12
      PRINT 1718
 1718 FORMAT(1H1,' INITIAL EXCITATIONS ... ELEMENT NUMBER WITH REAL AND
     1IMAGINARY PARTS OF EXCITATIONS')
      PRINT 1717,(I,COEF(I),I=1,N)
 1717 FORMAT(  4(I4,'=(',E12.7,',',E12.7,')',1X))
   12 NDIV1=NDIV-1
      NDIV2=NDIV-2
C
C     **************************************
C     BEGIN GROUP COORDINATE RELAXATION LOOP
C     **************************************
C
      DO 2000 LOOP=1,LOOPMX
      IF(IPRINT.EQ.0)GO TO 27
      CALL FIRST
   27 ITINDX=2
      DO 5 I=1,NRELAX
      DO 5 J=1,NRELAX
      AMAIN(I,J)=CMPLX(0.,0.)
    5 ASIDE(I,J)=CMPLX(0.,0.)
      DO  7 I=2,NRELAX
      INDX(I)=(LOOP-1)*NRELX1+I-1
      INDX(I)=INDX(I)-N*((INDX(I)-1)/N)
      II=INDX(I)
      DC(I)=COEF(II)
    7 COEF(II)=CMPLX(0.,0.)
C
C     COLLECT THE VERTICES OF THE ICOSAHEDRON
C
      CALL ARRAYS(NRELAX,AMAIN,ASIDE,X,Y,Z,DL,DIRECT,COEF,P,1,12,INDX,N)
      IF(NDIV.EQ.1)GO TO 1000
C
C     COLLECT THE EDGES OF THE ICOSAHEDRON
C
      DO 200 J=1,NDIV1
      K=NDIV-J
      Q(1,1)=J*P(1,1)+K*P(1,3)
      Q(2,1)=J*P(2,1)+K*P(2,3)
      Q(3,1)=J*P(3,1)+K*P(3,3)
      Q(1,2)=J*P(1,2)+K*P(1,3)
      Q(2,2)=J*P(2,2)+K*P(2,3)
      Q(3,2)=J*P(3,2)+K*P(3,3)
      Q(1,3)=J*P(1,4)+K*P(1,3)
      Q(2,3)=J*P(2,4)+K*P(2,3)
      Q(3,3)=J*P(3,4)+K*P(3,3)
      Q(1,4)=J*P(1,4)+K*P(1,2)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      Q(2,4)=J*P(2,4)+K*P(2,2)
      Q(3,4)=J*P(3,4)+K*P(3,2)
      Q(1,5)=J*P(1,4)+K*P(1,5)
      Q(2,5)=J*P(2,4)+K*P(2,5)
      Q(3,5)=J*P(3,4)+K*P(3,5)
      Q(1,6)=J*P(1,4)+K*P(1,6)
      Q(2,6)=J*P(2,4)+K*P(2,6)
      Q(3,6)=J*P(3,4)+K*P(3,6)
      CALL ARRAYS(NRELAX,AMAIN,ASIDE,X,Y,Z,DL,DIRECT,COEF,Q,5,6,INDX,N)
  200 CONTINUE
      IF(NDIV.EQ.2)GO TO 1000
C
C     COLLECT THE FACES OF THE ICOSAHEDRON
C
      DO 300 J=1,NDIV2
      DO 300 K=1,NDIV2
      IF((J+K).GT.NDIV1)GO TO 300
      L=NDIV-J-K
      Q(1,1)=J*P(1,1)+K*P(1,2)+L*P(1,3)
      Q(2,1)=J*P(2,1)+K*P(2,2)+L*P(2,3)
      Q(3,1)=J*P(3,1)+K*P(3,2)+L*P(3,3)
      Q(1,2)=J*P(1,2)+K*P(1,3)+L*P(1,4)
      Q(2,2)=J*P(2,2)+K*P(2,3)+L*P(2,4)
      Q(3,2)=J*P(3,2)+K*P(3,3)+L*P(3,4)
      Q(1,3)=J*P(1,2)+K*P(1,5)+L*P(1,4)
      Q(2,3)=J*P(2,2)+K*P(2,5)+L*P(2,4)
      Q(3,3)=J*P(3,2)+K*P(3,5)+L*P(3,4)
      Q(1,4)=J*P(1,4)+K*P(1,5)+L*P(1,6)
      Q(2,4)=J*P(2,4)+K*P(2,5)+L*P(2,6)
      Q(3,4)=J*P(3,4)+K*P(3,5)+L*P(3,6)
      CALL ARRAYS(NRELAX,AMAIN,ASIDE,X,Y,Z,DL,DIRECT,COEF,Q,5,4,INDX,N)
  300 CONTINUE
C
C     POLISH THE MATRICES
C
 1000 ITINDX=1
      IF(INDEX.NE.1)GO TO 1002
      Q(1,1)=ALPHAD
      Q(2,1)=BETAD
      Q(3,1)=GAMMAD
      CALL ARRAYS(NRELAX,AMAIN,ASIDE,X,Y,Z,DL,DIRECT,COEF,Q,1,1,INDX,N)
 1002 DO 1001 I=1,NRELAX
      DUM=REAL(AMAIN(I,I))
      AMAIN(I,I)=CMPLX(DUM,0.0)
      DUM=REAL(ASIDE(I,I))
      ASIDE(I,I)=CMPLX(DUM,0.0)
      DO 1001 J=I,NRELAX
      ASIDE(I,J)=ASIDE(I,J)+AMAIN(I,J)
      AMAIN(J,I)=CONJG(AMAIN(I,J))
 1001 ASIDE(J,I)=CONJG(ASIDE(I,J))
C
C     DO THE EIGENPROBLEM FOR THE GROUP OF COORDINATES
C
      CALL FINISH(COEF,RATIO,NRELAX,AMAIN,ASIDE,V,VX,D,WK,INDX,N)
      IF(WK(1).GT.0..AND.VSAVE.LE.WK(1))GO TO 1999
      DO 1998 I=2,NRELAX
      II=INDX(I)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
 1998 COEF(II)=DC(I)
      GO TO 2001
 1999 IF(IPRINT.EQ.0)GO TO 1997
      CALL SECOND(HTIME)
      PRINT 1719,WK(1),LOOP,HTIME
 1719 FORMAT(//' NEW EXCITATIONS ...LARGEST EIGENVALUE = ',E14.9,' ...'
     1,'RELAXATION ITERATION NUMBER = ',I5,' ...ELAPSED TIME = ',2A6,
     2'SEC')
      DO 1720 I=2,NRELAX
      II=INDX(I)
 1720 DIRECT(I)=COEF(II)
      IF(IPUNCH.NE.1)GO TO 1722
      PUNCH 1723,(INDX(I),DIRECT(I),I=2,NRELAX)
 1723 FORMAT(3(I4,2E11.6))
 1722 PRINT 1717,(INDX(I),DIRECT(I),I=2,NRELAX)
 1997 IF(ABS(1.-VSAVE/WK(1)).LE.ABS(EPSI))GO TO 2010
 2000 VSAVE=WK(1)
C
C     ************************************
C     END GROUP COORDINATE RELAXATION LOOP
C     ************************************
C
      IF(LOOPMX.EQ.1)GO TO 2010
      PRINT 2003
 2003 FORMAT(' MAXIMUM NUMBER OF RELAXATIONS COMPLETED WITHOUT CONVERGEN
     1CE.')
      GO TO 2010
 2001 PRINT 2002
 2002 FORMAT(' THE GROUP COORDINATE RELAXATION METHOD HAS STOPPED YIELDI
     1NG STEADILY IMPROVED RESULTS. CONVERGENCE IS ASSUMED.')
 2010 LOOPMX=LOOP
      IF(N.NE.NRELAX)GO TO 157
      DO 170 K=1,N
  170 DC(K)=COEF(K)
  157 ITCOEF=1
      ITJC=INDX(1)
   57 RAT=10.*ALOG10(RATIO*V(ITJC))
      PRINT 100,RAT,LOOPMX,V(ITJC)
  100 FORMAT(1H1,/,' GENERALIZED DI (DB)    ',10X,'=',F20.9,/,' NUMBER OF
     1 RELAXATION ITERATIONS =    ',I17,/,' TRUE EIGENVALUE',17X,'=',
     2F20.9/)
      S=0.
      DO 45 K=1,N
   45 S=S+REAL(COEF(K))**2+AIMAG(COEF(K))**2
      IF(S.GT.1.E-20)GO TO 44
      DO 40 I=1,N
   40 DIRECT(I)=CMPLX(0.,0.)
      GO TO 103
   44 S=SQRT(S)
      DO 50 K=1,N
      COEF(K)=COEF(K)/S
      AMAG=CABS(COEF(K))
      IF(AMAG.LE.1.E-37)GO TO 49
      APHA=ATAN2(AIMAG(COEF(K)),REAL(COEF(K)))
      GO TO 50
   49 APHA=0.
   50 DIRECT(K)=CMPLX(AMAG,APHA)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
  103 PRINT 102
  102 FORMAT(' OPTIMUM ELEMENT EXCITATIONS ARE',/,'      ELT. NO.',18X,
     1'MAGNITUDE',23X,'PHASE')
      PRINT 101,(K,DIRECT(K),K=1,N)
  101 FORMAT(I10,2F30.9)
      IF(ITCOEF.EQ.ICOEF)GO TO 58
      ITJC=INDX(1)-ITCOEF
      IF(ITJC.LE.0)GO TO 58
      ITCOEF=ITCOEF+1
      VMAX=ABS(REAL(VX(1,ITJC)))+ABS(AIMAG(VX(1,ITJC)))
      IF(VMAX.GT.1.E-25)GO TO 150
      DO 131 J=2,NRELAX
      JJ=INDX(J)
  131 COEF(JJ)=VX(J,ITJC)
      COEF(N)=VX(1,ITJC)
      GO TO 57
  150 DO 160 J=2,NRELAX
      JJ=INDX(J)
  160 COEF(JJ)=VX(J,ITJC)/VX(1,ITJC)
      COEF(N)=CMPLX(1.,0.)
      GO TO 57
   58 ITJC=INDX(1)
      RATIO=RATIO*V(ITJC)
      IF(ICOFF.EQ.1)GO TO 59
      DO 60 K=1,N
   60 COEF(K)=DC(K)
   59 RETURN
      END

@PRT,S 3.ARRAYS
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
STREIT*RELAX(1).ARRAYS
C     SUBROUTINE ARRAYS WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
C     CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
C
C      SUBROUTINE ARRAYS FILLS THE TWO MATRICES FOR THE EIGENPROBLEM
C
C      NRELAX = 1 + THE NUMBER OF ELEMENT CURRENTS TO 'RELAX'
C      AMAIN = MATRIX OF QUADRATIC FORM FOR MAINLOBE REGION
C      ASIDE = MATRIX OF QUADRATIC FORM FOR SIDELOBE REGION
C      X(I) = X COORDINATE OF I-TH ELEMENT
C      Y(I) = Y COORDINATE OF I-TH ELEMENT
C      Z(I) = Z COORDINATE OF I-TH ELEMENT
C      DL = WORK AREA, HERE
C      DIRECT = WORK AREA, HERE
C      COEF = PRESENT VECTOR OF ELEMENT CURRENTS
C      Q = ARRAY OF POINTS  TO BE PROJECTED ONTO THE SPHERE
C      L1,L2 = LOOP LIMITS DEPENDING UPON WHERE THE POINTS Q LIE
C      INDX = INDICES OF ELEMENTS RELAXED IN THIS STEP
C      N = TOTAL NUMBER OF ELEMENTS IN THE ARRAY
C
      SUBROUTINE ARRAYS(NRELAX,AMAIN,ASIDE,X,Y,Z,DL,DIRECT,COEF,Q,L1,L2,
     1                  INDX,N)
C
      DIMENSION X(1),Y(1),Z(1),DL(1),Q(3,12),COSROT(5),SINROT(5),INDX(1)
      COMPLEX AMAIN(NRELAX,1),ASIDE(NRELAX,1),DIRECT(1),WTC,COEF(1)
      COMMON INOISE,IPHONE,WAVLEN,INDEX,ALPHAD,BETAD,GAMMAD,ICOFF,
     1       ISTEER,ALPHAS,BETAS,GAMMAS,NDIV,LOOPMX,EPSI,IPRINT,IPUNCH
      COMMON/DIR/IDIR1,IDIR2
      COMMON/EXTRA/PI2LAM,ITINDX
      DOUBLE PRECISION DUM
      DATA COSROT/1.,.309016994,-.809016994,-.809016994,.309016994/
      DATA SINROT/0.,.951056516,.587785252,-.587785252,-.951056516/
      DATA IDIR1,IDIR2/0,0/
      DATA EPSI2/-1.E-6/
      DO 3000 L=1,L2
      W=1./SQRT(Q(1,L)*Q(1,L)+Q(2,L)*Q(2,L)+Q(3,L)*Q(3,L))
      ALPH1=Q(1,L)*W
      BETA1=Q(2,L)*W
      GAMMA=Q(3,L)*W
C
C     ROTATE LOWER HALF SPHERE FOR SYMMETRY REASONS
      IF(GAMMA.GT.EPSI2)GO TO 30
       WT=-ALPH1*COSROT(4)+BETA1*SINROT(4)
      BETA1=-BETA1*COSROT(4)-ALPH1*SINROT(4)
      ALPH1=WT
C
   30 DO 3000 M=1,L1
      A=ALPH1*COSROT(M)-BETA1*SINROT(M)
      B=BETA1*COSROT(M)+ALPH1*SINROT(M)
      C=GAMMA
      ALPHA=A
      BETA =B
      IDIR=ITINDX
      IF(INDEX.EQ.1)GO TO 4
      CALL  REGION(A,B,C,IDIR)
      IF(IDIR.GE.1.AND.IDIR.LE.3)GO TO 5
      PRINT 1,IDIR,ALPHA,BETA,GAMMA
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      1 FORMAT(//' ****** SUBROUTINE REGION GIVES AN ILLEGAL RETURN.'/
       1         ' IDIR = ',I7,' FOR DIRECTION COSINES ALPHA = ',
       2           E15.8,' BETA = ',E15.8,' GAMMA = ',E15.8)
        GO TO 3000
      5 IF(IDIR.EQ.3)GO TO 3000
      4 IF(IDIR.EQ.2)IDIR2=IDIR2+1
        IF(IDIR.EQ.1)IDIR1=IDIR1+1
        WT=1.
        IF(INOISE.EQ.0)GO TO 10
        A=ALPHA
        B=BETA
        C=GAMMA
        WT=DNOISE(A,B,C)
        IF(WT.GE.0.)GO TO 9
        PRINT 7,WT,ALPHA,BETA,GAMMA
      7 FORMAT(//' ****** SUBROUTINE DNOISE GIVES AN ILLEGAL RETURN.'/
       1   ' DNOISE = ',E15.8,' FOR DIRECTION COSINES ALPHA = ',
       2           E15.8,' BETA = ',E15.8,' GAMMA = ',E15.8)
        GO TO 3000
      9 WT=SQRT(WT)
     10 IF(IPHONE.EQ.0)GO TO 20
        A=ALPHA
        B=BETA
        C=GAMMA
        I=N
        CALL PHONE(A,B,C,DIRECT,I,X,Y,Z)
        GO TO 21
     20 DO 11 I=1,N
     11 DIRECT(I)=CMPLX(1.0,0.0)
     21 WTC=CMPLX(0.,0.)
        DO 25 I=1,N
        DUM=(X(I)*ALPHA+Y(I)*BETA+Z(I)*GAMMA-DL(I))*PI2LAM
        A=COS(DUM)*WT
        B=SIN(DUM)*WT
        DIRECT(I)=CMPLX(A,B)*DIRECT(I)
     25 WTC=WTC+COEF(I)*DIRECT(I)
        WTC2=REAL(WTC)*REAL(WTC) + AIMAG(WTC)*AIMAG(WTC)
        WTC=CONJG(WTC)
        IF(IDIR.EQ.2)GO TO 2000
        AMAIN(1,1)=AMAIN(1,1)+WTC2
        DO 1000 I=2,NRELAX
        II=INDX(I)
        AMAIN(1,I)=AMAIN(1,I)+DIRECT(II)*WTC
        DO 1000 J=I,NRELAX
        JJ=INDX(J)
   1000 AMAIN(I,J)=AMAIN(I,J)+CONJG(DIRECT(II))*DIRECT(JJ)
        GO TO 3000
   2000 ASIDE(1,1)=ASIDE(1,1)+WTC2
        DO 2001 I=2,NRELAX
        II=INDX(I)
        ASIDE(1,I)=ASIDE(1,I)+DIRECT(II)*WTC
        DO 2001 J=I,NRELAX
        JJ=INDX(J)
   2001 ASIDE(I,J)=ASIDE(I,J)+CONJG(DIRECT(II))*DIRECT(JJ)
   3000 CONTINUE
        RETURN
        END
```

```
@PRT,S N.FINISH
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
STREIT*RELAX(1).FINISH
      C      SUBROUTINE FINISH WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
      C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
      C
      C       SUBROUTINE FINISH SOLVES THE EIGENPROBLEM AND RETURNS THE COEFFICIENTS
      C
      C       COEF(I) = THE EXCITATION COEFFICIENT OF THE I-TH ELEMENT
      C       RATIO = ESTIMATE OF THE RATIO OF POWER IN THE MAINLOBE REGION TO
      C               POWER IN THE SIDELOBE REGION. RATIO IS NOT IN DB.
      C       NRELAX = NUMBER OF RELAXED ELEMENTS + 1
      C       AMAIN = MATRIX OF QUADRATIC FORM FOR MAINLOBE REGION
      C       ASIDE = MATRIX OF QUADRATIC FORM FOR SIDELOBE REGION
      C       INDX = INDICES OF ELEMENTS RELAXED IN THIS STEP
      C       N = NUMBER OF ELEMENTS IN THE ARRAY
      C       THE REMAINDER OF THE ARGUMENTS ARE AS REQUIRED FOR SUBROUTINE PENCLH
      C
             SUBROUTINE FINISH(COEF,RATIO,NRELAX,AMAIN,ASIDE,V,VX,D,WK,INDX,N)
      C
             DIMENSION V(1),D(1),WK(1),INDX(1)
             COMPLEX AMAIN(NRELAX,1),ASIDE(NRELAX,1),VX(NRELAX,1),COEF(1)
             COMMON/DIR/IDIR1,IDIR2
             M=NRELAX
             NOPT=1
             CALL PENCLH(AMAIN,ASIDE,M,M,V,VX,M,NOPT,D ,WK,$1000)
             JJ=WK(2)
             IF(JJ.EQ.0)GO TO 1000
             IF(JJ.EQ.NRELAX)GO TO 15
             JJJ=NRELAX-JJ
             PRINT 2001,JJJ
        2001 FORMAT(///' XXXXX  NUMERICAL DIFFICULTIES IN THE EIGENANALYSIS PRE
            1VENT THE COMPLETE SOLUTION OF YOUR PROBLEM.'/
            2 ' XXXXX  THE BEST OF THE COMPUTED COEFFICIENTS IS RETURNED.'/
            3' XXXXX  THERE ARE AT MOST ',I5,' LINEARLY INDEPENDENT BETTER SETS
            4 OF COEFFICIENTS.'///)
          15 IT=1
             VMAX=-1.E+20
             DO 20 J=1,JJ
             IF(V(J).LT.VMAX)GO TO 20
             VMAX=V(J)
             IT=J
          20 CONTINUE
             INDX(1)=IT
             WK(1)=VMAX
             RATIO=      FLOAT(IDIR2)/FLOAT(IDIR1)
             VMAX=ABS(REAL(VX(1,IT)))+ABS(AIMAG(VX(1,IT)))
             IF(VMAX.GT.1.E-25)GO TO 50
             DO 30 J=1,N
          30 COEF(J)=COEF(J)*VX(1,IT)
             DO 31 J=2,NRELAX
             JJ=INDX(J)
          31 COEF(JJ)=VX(J,IT)
             GO TO 1002
          50 DO 60 J=2,NRELAX
             JJ=INDX(J)
          60 COEF(JJ)=VX(J,IT)/VX(1,IT)
             GO TO 1002
        1000 PRINT 1001
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      1001 FORMAT(///' XXXXX  NUMERICAL DIFFICULTIES IN THE EIGENANALYSIS PRE
          1VENT THE SOLUTION OF YOUR PROBLEM, SORRY, CHARLEY.'///)
           WK(1)=-1.
      1002 IDIR1=0
           IDIR2=0
           RETURN
           END

@PRT,S H.PENCLH
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
STREIT*RELAX(1).PENCLH
      C      SUBROUTINE PENCLH WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
      C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   17 MAY 1976
      C
      C       SUBROUTINE PENCLH COMPUTES ALL THE EIGENVALUES AND ALL THE
      C       EIGENVECTORS OF THE GENERALIZED EIGENPROBLEM
      C                          A * Z = LAMBDA * B * Z
      C       WHERE A AND B ARE HERMITIAN MATRICES AND B IS POSITIVE DEFINITE.
      C
      C       THIS ROUTINE USES TWO PROGRAMS OF  IMSL (INTERNATIONAL MATHEMATICAL
      C       AND STATISTICAL LIBRARIES,INC., HOUSTON, TEXAS) FOR THE SOLUTION OF
      C       THE ORDINARY  HERMITIAN EIGENPROBLEM (THAT IS, WHEN THE HERMITIAN
      C       MATRIX B IS JUST THE IDENTITY MATRIX). ANY SET OF ROUTINES FOR SOLVING
      C       THIS PROBLEM CAN BE SUBSTITUTED IF CARE IS USED.
      C
      C       THIS ROUTINE EXTENDS TO THE HERMITIAN CASE THE ALGORITHM OF
      C       R.S. MARTIN AND J.H. WILKINSON, 'REDUCTION OF THE SYMMETRIC EIGEN-
      C       PROBLEM A*X = LAMBDA B*X      AND RELATED PROBLEMS TO STANDARD FORM',
      C       NUMER. MATH., VOL. 11, P.99-110, 1968, WHICH IS DESIGNED FOR THE REAL
      C       CASE.
      C
      C    CAUTION *** ALL OF A AND THE STRICT LOWER TRIANGLE OF B ARE DESTROYED.
      C
      C       A,B  = HERMITIAN MATRICES OF THE EIGENPROBLEM. COMPLEX MATRICES
      C                DIMENSIONED  IN X IN.
      C       N = THE ORDER OF THE EIGENPROBLEM
      C       IN = THE ACTUAL ROW DIMENSION OF BOTH A AND B.
      C       VALUE = WILL CONTAIN THE EIGENVALUES. REAL ARRAY DIMENSIONED AT LEAST N.
      C       VECTOR = WILL CONTAIN THE EIGENVECTORS. COLUMN J OF VECTOR CONTAINS
      C                THE EIGENVECTOR CORRESPONDING TO THE EIGENVALUE VALUE(J).
      C                COMPLEX ARRAY DIMENSIONED IV X N.
      C       IV = ROW DIMENSION OF VECTOR. MUST HAVE IV.GE.N
      C       NOPT = OPTION INDICATOR.
      C            = 0 , COMPUTE EIGENVALUES ONLY.
      C            = 1 , COMPUTE EIGENVALUES AND EIGENVECTORS.
      C       DL = REAL ARRAY DIMENSIONED AT LEAST N. WORK AREA.
      C       WK = REAL ARRAY DIMENSIONED AT LEAST 3*N. WORK AREA WHOSE CONTENTS
      C            ARE DESTROYED. ON NORMAL RETURN, WK(2) CONTAINS THE NUMBER OF
      C            CORRECTLY COMPUTED EIGENVECTORS. ERROR RETURN, WK(2) = -1.
      C       $ = A STATEMENT NUMBER IN THE CALLING PROGRAM. RETURN TO STATEMENT $
      C           ONLY IF B IS NOT POSITIVE DEFINITE. (CAN BE CAUSED BY ROUNDING ERROR)
      C
             SUBROUTINE PENCLH(A,B,N,IN,VALUE,VECTOR,IV,NOPT,DL,WK,$)
      C
             DIMENSION A(IN,1),B(IN,1),VALUE(1),VECTOR(IV,1),DL(1),WK(1)
             COMPLEX A,B,VECTOR,X
             DOUBLE PRECISION Z
      C      BEGIN MARTIN AND WILKINSON ROUTINE REDUC1
             IF(N.GT.0)GO TO 5
             N=-N
             GO TO 11
           5 WK(2)=-1.
             DO 1 I=1,N
             DO 1 J=1,N
             X=CONJG(B(I,J))
             IF(I.EQ.1)GO TO 4
             IP=I-1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      DO 3 K=IP,1,-1
    3 X=X-CONJG(B(I,K))*B(J,K)
    4 IF(I.NE.J)GO TO 2
      IF(REAL(X).LE.0.)RETURN 11
      IF(ABS(AIMAG(X)/REAL(X)).GE.1.E-4)RETURN 11
      Z=REAL(X)**2
      Z=Z+AIMAG(X)**2
      DL(I)=SQRT(SQRT(Z))
      GO TO 1
    2 B(J,I)=X/DL(I)
    1 CONTINUE
   11 DO 20 I=1,N
      DO 20 J=I,N
      X=A(I,J)
      IF(I.EQ.1)GO TO 14
      IP=I-1
      DO 16 K=IP,1,-1
   16 X=X-B(I,K)*A(J,K)
   14 A(J,I)=X/DL(I)
   20 CONTINUE
      DO 30 J=1,N
      DO 30 I=J,N
      X=A(I,J)
      IF(I.EQ.J)GO TO 44
      IP=I-1
      DO 35 K=IP,J,-1
   35 X=X-A(K,J)*CONJG(B(I,K))
   44 IF(J.EQ.1)GO TO 45
      IP=J-1
      DO 36 K=IP,1,-1
   36 X=X-CONJG(A(J,K)*B(I,K))
   45 A(I,J)=X/DL(I)
   30 CONTINUE
C     FIND THE EIGENVALUES/EIGENVECTORS
      DO 50 I=1,N
      Y=REAL(A(I,I))
      A(I,I)=CMPLX(Y,0.)
      DO 50 J=I,N
      A(I,J)=A(J,I)
   50 A(J,I)=CONJG(A(J,I))
C     IMSL ROUTINE
      CALL VCVTCH(A,N,IN,A)
      IF(NOPT.LT.0.OR.NOPT.GT.2)NOPT=1
      IF(NOPT.NE.0.AND.IV.LT.N)NOPT=0
C     IMSL ROUTINE
      CALL EIGCH(A   ,N,NOPT,VALUE,VECTOR,IV,WK,IER)
      M2=N
      IF(IER.GE.128)M2=IER-128
      IF(M2.GE.1)GO TO 51
      WK(2)=0.
      RETURN
   51 WK(2)=M2
      IF(NOPT.EQ.0.OR.NOPT.EQ.3)RETURN
      DO 90 J=1,M2
      Z=0.
      DO 80 I=1,N
   80 Z=Z+REAL(VECTOR(I,J))**2+AIMAG(VECTOR(I,J))**2
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
          Y=SQRT(Z)
          DO 81 I=1,N
       81 VFCTOR(I,J)=VECTOR(I,J)/Y
       90 CONTINUE
C         BEGIN MARTIN AND WILKINSON ROUTINF REBAKA
          DO 60 J=1,M2
          DO 60 I=N,1,-1
          X=VECTOR(I,J)
          IF(I.EQ.N)GO TO 71
          1P=I+1
          DO 70 K=1P,N
       70 X=X-CONJG(B(K,I))*VECTOR(K,J)
       71 VFCTOR(I,J)=X/DL(I)
       60 CONTINUE
          RETURN
          END

@PRT,S B.REGION
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
STREIT*RELAX(1).REGION
C     SUBROUTINE REGION WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
C     CENTER, NEW LONDON LABORATORY, NEW LONDON, CT    10 AUGUST 1977
C
C      SUBROUTINE REGION DIVIDES POLAR SPACE INTO THREE DISJOINT REGIONS
C
C      ALPHA = DIRECTION COSINE WITH RESPECT TO THE X-AXIS
C      BETA  = DIRECTION COSINE WITH RESPECT TO THE Y-AXIS
C      GAMMA = DIRECTION COSINE WITH RESPECT TO THE Z-AXIS
C      IDIR  = 1  IF DIRECTION COSINES LIE IN THE MAINLOBE REGION
C            = 2  IF DIRECTION COSINES LIE IN THE SIDELOBE REGION
C            = 3  IF DIRECTION COSINES LIE IN THE IGNORED REGION
C
      SUBROUTINE REGION(ALPHA,BETA,GAMMA,IDIR)
C
C     *************************************
C     BEGIN CODE FOR SPECIFIC ARRAY PROBLEM
C     *************************************
C
C     SET COSINE OF 82 DEGREES
      DATA COS82/.1391731009/
      IDIR=2
      IF(ABS(BETA).LT.COS82)IDIR=1
C
C     ***********************************
C     END CODE FOR SPECIFIC ARRAY PROBLEM
C     ***********************************
C
      RETURN
      END
```

```
STREIT*RELAX(1).DNOISE
C      SUBROUTINE DNOISE WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
C
C       FUNCTION DNOISE COMPUTES THE NOISE LEVEL IN ANY DIRECTION
C
C       ALPHA = DIRECTION COSINE WITH RESPECT TO THE X-AXIS
C       BETA  = DIRECTION COSINE WITH RESPECT TO THE Y-AXIS
C       GAMMA = DIRECTION COSINE WITH RESPECT TO THE Z-AXIS
C       DNOISE= NOISE LEVEL IN THE SPECIFIED DIRECTION
C
       FUNCTION DNOISE(ALPHA,BETA,GAMMA)
C
C      **************************************
C      BEGIN CODE FOR SPECIFIC ARRAY PROBLEM
C      **************************************
C
       DNOISE=1.
C
C      ************************************
C      END CODE FOR SPECIFIC ARRAY PROBLEM
C      ************************************
C
       RETURN
       END

*PRT,S B.PHONE
```

```
STREIT*RELAX(1).PHONE
      C      SUBROUTINE PHONE  WRITTEN BY  ROY L. STREIT, NAVAL UNDERWATER SYSTEMS
      C      CENTER, NEW LONDON LABORATORY, NEW LONDON, CT   10 AUGUST 1977
      C
      C       SUBROUTINE PHONE COMPUTES THE RESPONSE PATTERN OF EACH HYDROPHONE.
      C       THE USER MUST TAKE CARE NOT TO CHANGE THE X, Y, AND Z ARRAYS.
      C
      C       ALPHA = DIRECTION COSINE WITH RESPECT TO THE X-AXIS
      C       BETA  = DIRECTION COSINE WITH RESPECT TO THE Y-AXIS
      C       GAMMA = DIRECTION COSINE WITH RESPECT TO THE Z-AXIS
      C       DIRECT(1),...., DIRECT(N) = THE DIRECTIONAL RESPONSE OF N HYDROPHONES
      C                                   IN THE SPECIFIED DIRECTION.
      C                                   (NOTE THAT THE RESPONSE MAY BE COMPLEX)
      C       N = TOTAL NUMBER OF HYDROPHONES
      C       X(I) = X COORDINATE OF I-TH HYDROPHONE
      C       Y(I) = Y COORDINATE OF I-TH HYDROPHONE
      C       Z(I) = Z COORDINATE OF I-TH HYDROPHONE
      C
             SUBROUTINE PHONE(ALPHA,BETA,GAMMA,DIRECT,N,X,Y,Z)
      C
             COMPLEX DIRECT(1)
             DIMENSION X(1),Y(1),Z(1)
      C
      C      ****************************************
      C      BEGIN CODE FOR SPECIFIC ARRAY PROBLEM
      C      ****************************************
      C
             DO 10 I=1,N
          10 DIRECT(I)=CMPLX(1.0,0.0)
      C
      C      ************************************
      C      END CODE FOR SPECIFIC ARRAY PROBLEM
      C      ************************************
      C
             RETURN
             END

@ FIN
```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
NORDA (Dr. D. J. Ramsdale)	1
NAVSEASYSCOM, SEA-09G32,	1
NAVSEASYSCOMHQ, Code 03421 (E. Liszka)	1
NAVOCEANSYSCEN (Dr. F. J. Harris)	1
NAVPGSCOL	1
ARL/PENN STATE, STATE COLLEGE (Dr. W. Thompson, Dr. Geoffrey Wilson)	2
DDC, ALEXANDRIA	12
Engineering Societies Library, United Engineering Center	1
Dr. C. J. Moore, Rolls-Royce Limited, Derby Engine Division, Advanced Research Laboratory, P.O. Box 31, Derby, England	1
Dr. G. Charles, Documentation C.E.R.T.S.M., D.C.A.N. Toulon, F 83800 Toulan-Naval, France	1
Prof. Mithat Idemen, Applied Mathematics Division, Marmara Research Institute, P.K. 141, Kadikoy Istanbul, Turkey	1
Capt. Harold Cox, 8112 Saxony Drive, Annandale, VA 22003	1
Dr. T. Kooij, ARPA Research Center, Unit 1, Moffett Field CA 94035	1
Dr. D. W. Tufts, Dept. of Electrical Engineering, University of Rhode Island, Kingston, RI 02881	1
Dr. J. T. Lewis, Mathematics Dept. University of Rhode Island, Kingston, RI 02881	1
Dr. M. T. Ma, Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Dept. of Commerce, Boulder, CO	1
Dr. H. S. C. Wang, E.D.O. Corporation, College Point, N.Y. 11356	1
Dr. R. N. McDonough, University of Delaware, Newark, DE 19711	1
Dr. J. S. Papadakis, Mathematics Department, University of Rhode Island, Kingston, RI 02881	1
Dr. R. C. Hansen, Box 215, Tarzana, CA 91356	1
Dr. Daniel Suchman, Sea Operations Department, General Motors Corp., Santa Barbara Operations, 6767 Hollister Avenue, Goleta, CA 93017	1
Dr. T. Sawataki, Bendix Research Laboratories, Southfield, MI 48076	1
CAPT Bohannan, 37 Vinegar Hill Road, Ledyard, CT	1
Dr. Bernard Steinberg, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104	1
Dr. David K. Cheng, Electrical and Computer Engineering Dept., Syracuse University, Syracuse, NY 13210	1
Dr. Allan D. Waren, Chairman, Computer and Information Science, The Cleveland State University, Cleveland, OH 44115	1
Mr. Anthony Rockmore, Systems Control, Inc., 1801 Page Mill Road, Palo Alto, CA 94304	1
Prof. Harold Ladd, Jr., Old Dominion Univ., Math Dept., Hampton Blvd. Norfolk, VA 23508	1
Mr. Bill Kesner, Bldg 302, Westinghouse Research Development Center, Beulah Road, Pittsburgh, PA 15235	1